

B-SPLINE FINITE ELEMENTS FOR PLANE ELASTICITY PROBLEMS

A Thesis

by

BHAVYA AGGARWAL

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE

December 2006

Major Subject: Aerospace Engineering

B-SPLINE FINITE ELEMENTS FOR PLANE ELASTICITY PROBLEMS

A Thesis

by

BHAVYA AGGARWAL

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

Chair of Committee, John. D. Whitcomb
Committee Member, Walter E. Haisler
Arun Srinivasa
Head of Department, Helen Reed

December 2006

Major Subject: Aerospace Engineering

ABSTRACT

B-spline Finite Elements for Plane Elasticity Problems. (December 2006)

Bhavya Aggarwal, B.S., Punjab Engineering College, India

Chair of Advisory Committee: Dr. John D. Whitcomb

The finite element method since its development in the 1950's has been used extensively in solving complex problems involving partial differential equations. The conventional finite element methods use piecewise Lagrange interpolation functions for approximating displacements. The aim of this research is to explore finite element analysis using B-spline interpolation. B-splines are piecewise defined polynomial curves which provide higher continuity of derivatives than piecewise Lagrange interpolation functions. This work focuses on the implementation and comparison of the B-spline finite elements in contrast with the conventional finite elements. This thesis observes that the use of B-spline interpolation functions can reduce the computational cost significantly. It is an efficient technique and can be conveniently implemented into the existing finite element programs.

To, my parents, for their love, faith and support

ACKNOWLEDGMENTS

This thesis marks the end of the beautiful journey to achieve my Master's degree in Aerospace Engineering. Throughout this journey I have been supported and guided by several people. I would like to take this opportunity to express my gratitude to all those people.

I would like to acknowledge the supervision and support of my advisor Dr. John D. Whitcomb for this work. His enthusiasm and patience have been very valuable in the completion of this work. His mission of producing high-quality work will always help me grow and expand my thinking.

My sincere thanks to Dr. Walter E. Haisler, Director of Graduate Program, for his guidance and support throughout my graduate studies and for serving on my committee. I would also like to thank Dr. Arun Srinivasa for serving on my committee and providing his valuable time and suggestions.

I would also like to acknowledge the help provided by the Aerospace Engineering Department staff, especially Ms. Karen Knabe, for their timely help in all the administrative work.

I would also like to gratefully acknowledge the support of my research group: Deepak Goyal, Julian Varghese, Jongil Lim, Jiwoong Sue, Brian Owens. Deepak helped me immensely by constant encouragement and friendship. My sincere thanks to Julian for his valuable advice and friendly help.

I also wish to thank my friends, Shalini, Deepak, Pranab, Akhilesh, Aradhana and Jyothasana, for all the love and care they have always provided and for emotionally supporting me through my tough times.

Finally, I am grateful to my parents and my sisters for their unconditional love and support in raising me to be the person I am today.

TABLE OF CONTENTS

	Page
ABSTRACT.....	iii
DEDICATION.....	iv
ACKNOWLEDGMENTS.....	v
TABLE OF CONTENTS	vi
LIST OF FIGURES.....	viii
CHAPTER	
I INTRODUCTION.....	1
1. Overview	1
2. Background	2
3. Objectives.....	5
II INTERPOLATION	7
1. Lagrange Interpolation	8
2. B-spline Interpolation.....	12
3. Summary of Properties of B-spline Basis Functions	15
4. Knot Vectors	16
5. B-Spline for Two Dimensions.....	22
III B-SPLINE FINITE ELEMENTS- 1D.....	24
1. Elastic Rod	29
IV B-SPLINE FINITE ELEMENTS- 2D.....	33
1. Plane Elasticity	33
2. Weak Form.....	35
3. Discretized Weak Form.....	35
4. Sub-Parametric Formulation	37
5. Global Derivatives.....	38
6. Differential Area	39
7. B-spline Finite Elements	39
8. Assembly of Stiffness Matrix.....	45
V COMPARISON OF THE B-SPLINE ELEMENTS WITH SERENDIPITY	
ELEMENTS	49

CHAPTER	Page
1. Single Lap Shear Joint.....	49
2. Plate with Elliptical Hole	64
VI CONCLUSIONS AND FUTURE WORK	74
REFERENCES	76
VITA	80

LIST OF FIGURES

FIGURE	Page
1 Polynomial interpolation of degree 6 to interpolate 7 points	8
2 Quadratic Lagrange interpolation functions.....	9
3 Curve obtained by interpolating a set of three points using quadratic Lagrange shape functions.....	10
4 The effect of change in one coordinate in the interpolation of a set of data points with the change in interpolation type.....	11
5 Piecewise defined cubic B-Spline basis function.....	14
6 Uniform cubic basis functions.....	14
7 Local support interval for linear and quadratic spline functions.....	15
8 Effect of knot spacing on shape of basis functions	17
9 Periodic splines using uniform knot vectors	18
10 Open uniform quadratic basis functions for knot vector $[0,0,0,1,2,3,4,4,4]$...	19
11 Open uniform cubic basis functions for knot vector $[0,0,0,0,1,1,1,1]$	20
12 Quadratic basis functions for a non-uniform knot vector $[0, 0, 0, 0.5, 1, 1.5, 2, 2.5, 2.5]$	20
13 Comparison of B-spline curves for control points $[[0, 0], [4,9], [6,10], [8,5]]$ using three different knot vectors	21
14 B-spline patches for two dimensions	22
15 Two cubic Lagrange elements.....	24
16 Open uniform B-spline basis functions of cubic order defined for each cluster	25
17 Integration domains for B-spline finite elements	27
18 B-spline basis functions for different knot intervals	28
19 Basis functions for a cluster with single patch.....	28
20 Transformation to normalized coordinates	30
21 Stiffness matrix assembly for a cluster for 1D analysis	32

FIGURE	Page
22 Three coordinate system for a 2D B-spline cluster	41
23 Coordinate mapping from physical coordinates to knot coordinates	43
24 Transformation from knot coordinates to local coordinates for each patch in a cluster	43
25 Renumbering of the control points for a cluster. $n_2 = p_2 + 4$ marked patch shows numbering for patch in the c^{th} column and r^{th} row.	46
26 Control point map for a patch	47
27 Control points numbering for a cluster with 8 control points on each side t_1 and t_2	48
28 Lap shear test configuration	50
29 Single lap joint configuration	51
30 Cluster mesh for B-spline finite elements	52
31 Deformed shape for the lap shear specimen.....	52
32 Contour plots of σ_{11} , σ_{22} , σ_{12} obtained by 8 node serendipity elements.	53
33 Variation of peel stress at the interface between the adhesive layer and the adherends	54
34 Variation of shear stress at the interface between the adhesive layer and the adherends	54
35 Convergence behavior for peel stress near the interface for conventional finite element solution	55
36 Convergence behavior for peel stress near the interface using B-spline elements	56
37 Difference in σ_{22} value with respect to the reference value for conventional and B-spline finite elements.	58
38 Solution time vs number of degrees of freedom for conventional and B-spline finite elements	59
39 Error in σ_{22} vs time for both conventional and B-spline finite elements.....	60

FIGURE	Page
40 Convergence behavior for shear stress near the interface using conventional finite element solution	61
41 Convergence behavior for shear stress near the interface using B-spline finite element solution	62
42 Difference in σ_{12} value with respect to the reference value for conventional and B-spline finite elements.	63
43 Error in σ_{12} vs time for both conventional and B-spline finite elements.....	64
44 Plate with elliptical hole.....	65
45 Deformed quarter plate with contours showing axial stress.....	66
46 Stress contours for σ_{22} for the reference solution.	66
47 Stress contours for σ_{12} for the reference solution.	67
48 Cluster mesh for B-spline finite element.....	68
49 Normalized difference in σ_{11} value with respect to the reference value for conventional finite elements.	68
50 Normalized difference in σ_{11} value with respect to the reference value for B-spline finite elements.	69
51 Solution time vs number of degrees of freedom for conventional and B-spline finite elements.....	70
52 Error in σ_{11} vs time for both conventional and B-spline finite elements.....	71
53 Normalized difference in σ_{22} value with respect to the reference value for conventional finite elements.	71
54 Normalized difference in σ_{22} value with respect to the reference value for B-spline finite elements.	72
55 Error in σ_{22} vs time for both conventional and B-spline finite elements.....	73

CHAPTER I

INTRODUCTION

1. Overview

Partial differential equations are used extensively in describing the physics of various phenomena in science. The physical laws of structural mechanics, fluid flow diffusion, electromagnetic fields, and quantum mechanics can all be written in the form of partial differential equations. Most often, numerical analysis techniques like finite difference and finite element are used to solve these partial differential equations for the unknowns. The development of finite element analysis began in the early 1950's to analyze structures in civil engineering and airframe analysis [1]. Since then, it has been used in a wide variety of applications to analyze complex systems.

For solving complex structural problems, finite element analysis makes use of various shape functions to interpolate using the unknown nodal displacements. Since the introduction of two-dimensional finite element analysis in 1956 [1], finite element analysis has always been an area of interest. The tool became a part of the normal product life cycle in the industry, with the advancement in computer technology. There have been continuous research efforts to improve the efficiency, flexibility and accuracy of the method.

In terms of interpolations used, Lagrange and Hermite type are the most popularly used interpolation functions [2]. Although, these interpolants are easy to use and can provide sufficient accuracy, they are computationally costly, especially for a more accurate and smooth solution. The aim of this research is to explore finite element analysis using B-spline interpolation. B-splines are piecewise defined polynomial curves. These interpolants provide higher smoothness, and are widely used in describing

This thesis follows the style and format of *Journal of Composite Materials*.

surfaces in CAD packages [3]. The intervals over which these piecewise functions are defined are called knot intervals.

There has been some work in the use of B-spline interpolation for solving partial differential equations [4], especially in the case of problems where continuity of higher derivatives is required [5]. It has been shown that these interpolation functions can result in a significant reduction of the number of degrees of freedom used in the analysis of different types of problems [6].

This work focuses on the implementation and comparison of the B-spline finite elements in contrast with the conventional finite elements. The implementation of B-spline elements for one-dimensional and two-dimensional elements in an existing finite element code will be discussed. A comparison of the B-spline finite element with the conventional method using Lagrange elements in terms of accuracy and efficiency will be performed for several plane elasticity problems.

2. Background

In the 1960's, to idealize the geometry of complex systems, B-spline basis functions were introduced [7]. The term 'spline' comes from the flexible spline devices used by shipbuilders and DRAFTSMEN to draw smooth shapes. The first reference to the word B-spline ('B' refers to Basis) in the field of mathematics was given by Schoenberg [7] in 1946, who described it as a smooth piecewise polynomial approximation. In the field of computer science, B-splines have been referred to as piecewise polynomial curves [8]. Since their introduction, they have been used as a tool to create smooth curves and surfaces in computer graphics. These functions provide a smooth interpolated curve for a large number of control points, and also provide a higher continuity of derivatives.

In 1962, [9] Carl De Boor's publication on faster and numerically stable algorithms for the calculation of spline interpolation functions marked the beginning of the use of splines in automobile modeling. B-splines were used to represent objects of the real world mathematically, which played an important part in the aircraft and automobile industry [10]. Now, not only are splines used for geometric design, they are also used in

the computation of flight trajectories [11]. B-Splines are now extensively used in the graphic design and CAD industry for creating smooth curves and surfaces.

Recently, a few studies have also been done in the use of B-spline interpolation functions to solve structural problems using finite element analysis. The main emphasis in the use of the B-Spline interpolation has been for analyzing plate and shell elements, or for integrating CAD with the mechanical analysis. The scope of the problems solved using B-spline interpolation has been very limited [6].

One of the earlier works in the use of B-spline basis functions for finite element method was done by Zamani [12]. The author used least squares finite element approximation of Poisson's equation and then subsequently for Navier's Equation [13]. It was concluded that the accuracy was good even for a coarse mesh.

B-splines provide a continuity of higher derivatives, which is a necessary condition for plate and shell elements. Hence, there has been considerable work in the application of B-spline interpolation for plate and shell elements [14-18]. The plate and shell elements require the continuity of derivatives, hence the use of B-spline elements simplifies the analysis and makes it much more efficient, as is observed by most authors.

The CAD geometric models are developed using B-spline interpolation, thus resulting in a smooth and accurate approximation of the actual surfaces. But, the finite element models are generally developed using Lagrange or Hermite interpolations, thus the data required for both the steps is different. The integration of CAD data with the finite element model is a major concern for an efficient design process.

The integration has been done in three ways. First is the use of the design elements for controlling the exact geometry, which are further refined into the regular elements. These design elements defined using B-spline interpolation have been used in the structural shape optimization of a finite element model [19, 20]. The finite elements are still defined by the regular interpolation functions. This laid the foundation for some of the later works to link the design phase with the finite element phase. The second method to integrate the CAD with the finite element model, is the generation of a finite element mesh using the geometric model data. A design process requires a constant data

exchange from the CAD phase to the finite element phase. The intermediate step requires mesh generation for the finite element model, which is a tedious process. B-spline interpolation has also been used to convert the geometric model data to the finite element mesh [21,22]. The author used B-splines for an automatic finite element mesh generation using the geometric model data. The third method of integration of design and analysis is by using B-spline elements for the approximation of the field variables instead of the Lagrange or Hermite type interpolation [23,24]. Cho and Roh [25], discuss the finite element method using B-spline for a shell integrating geometric design with the mechanical analysis. Subbarayan et al. [26] have proposed a similar methodology named as Constructive Solid Analysis analogous to Constructive Solid Geometry used for modeling. The procedure aims at integrating the design and analysis and also at enabling optimal shape design. Hughes et al. [27], have named the use of same interpolation functions for both design and analysis as isogeometric analysis, as it is an exact representation of geometry for the finite element model. The method uses the Non Uniform Rational B-spline commonly referred to as NURBS to create a coarse mesh from the geometric model. This coarse mesh is refined by a new strategy named as k-refinement. In the case of k-refinement, the order of the polynomials are elevated followed by a knot insertion. The k-refinement is reported to be much more efficient and robust than the standard h or p- refinement used in the conventional FEM models.

Besides the use of B-spline functions for structural problems, they have been used in some other fields as well. In [28], the author compared the conventional finite element using Lagrange type interpolation with the B-spline finite elements. The paper discusses the efficiency in solving Relativistic Mean Field Equations in terms of numerical expense, precision and convergence behavior. The paper reports a reduction in the numerical cost using B-spline FEM. The use of the B-spline finite element method for the thermistor problem [29,30] and for a numerical solution of Burger's equation[31-33] has been successfully developed and validated.

The focus of this thesis is on comparing the results from a regular Finite Element with B-spline FEM for 1D and 2D elasticity for accuracy and efficiency. The

implementation of the same in an existing finite element module will be discussed. This will give a direct comparison of the two methods in terms of the ease of implementation without changing the existing analysis tools already available. The model will be validated using various one and two dimensional plane elasticity problems. Parametric studies to compare the regular finite element model with the B-spline model will be conducted. The variation in the stress distribution with different models will be compared for accuracy and efficiency.

3. Objectives

The two main objectives of this research are the implementation of the B-Spline interpolation into an existing finite element model and to compare the results of the B-Spline finite element method with the conventional finite element method using one-dimensional and two-dimensional problems.

To accomplish these objectives, the basic differences in the Lagrange interpolation and B-Spline interpolation functions have been studied. A background on the understanding of the interpolation functions commonly used in surface design and modeling has been built. The existing finite element techniques with respect to performance efficiency, shape optimal design, and integration with CAD data has also been evaluated.

A strategy to implement the interpolation functions used in CAD modeling for a finite element model for one- dimensional and two-dimensional problems is discussed. A robust algorithm was created to implement the B-spline functions for a more efficient analysis. An existing in-house finite element code has been modified for the implementation of the new interpolation functions to analyze one- dimensional and two-dimensional structural problems.

The model is illustrated for one-dimensional and two-dimensional plane elasticity problems. The comparison of the new model with the existing conventional model is done using parametric studies for a few plane elasticity problems. The two models are compared in terms of the run-time and the number of degrees of freedom required to run

an efficient analysis. The accuracy of the model is validated with the results from the conventional finite element models, especially for the regions of stress concentrations.

This work gives an insight into the performance of the B-spline finite element model as opposed to the conventional models. It also provides a groundwork for a more efficient analysis tool with a higher compatibility with the CAD models.

After a brief introduction and history on B-splines, chapter II outlines the theory and concepts of the Lagrange interpolation and B-spline interpolation functions. Chapter III then describes how the B-spline functions can be implemented in a one-dimensional finite element code, setting a background for the steps involved in the 2D analysis. Chapter IV outlines in detail the steps involved in implementing the B-spline finite elements to a conventional finite element code. Finally chapter V discusses the performance of the two methods in terms of efficiency and accuracy with the help of some practical problems in plane elasticity.

CHAPTER II

INTERPOLATION

Interpolation is a method of estimating a continuously defined function between known values. In Finite Element Analysis (FEA), the known displacements are at the nodes. In conventional FEA, Lagrange and Hermite interpolation have been extensively used, due to its convenience in implementation. The collection of known values q_i corresponding to coordinate location x_i are called the control points.

There are different interpolation methods based on the need of accuracy, efficiency, computational cost, smoothness, etc. There are two main ways in which polynomial functions can be calculated to approximate a set of control points. In the first case the curve exactly passes through all the known control points and this method is called interpolation. In the second case, the curve may or may not exactly pass through the given control points, to fulfill other required conditions, like continuity or smoothness and is termed as approximation. The polynomial interpolation exactly passes through the control points, whereas the B-spline interpolation is a type of approximation. In general, the B-spline is also referred to as interpolation.

For a known set of n control points (x_j, q_j) , the interpolating function $P(x)$ is defined by:

$$P(x) = \sum_{j=1}^n B_j(x) q_j \quad 2.1$$

The interpolating polynomial $P(x)$ satisfies the condition in the case of most interpolation methods:

$$P(x_i) = q_i \quad \text{for } 1 \leq i \leq n \quad 2.2$$

For the case of B-splines, in order to maintain the continuity of the higher derivatives, this condition may or may not hold true. This will be discussed in the subsequent sections.

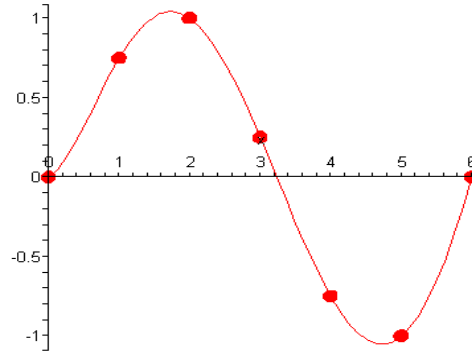


Fig.1. Polynomial interpolation of degree 6 to interpolate 7 points.

As shown in Fig.1, a polynomial for a curve which goes through all the seven control points is fitted. In conventional FEM, the polynomial interpolation is used in Lagrange or Hermite form of different degree, depending on the type of problem. The Lagrange type of interpolation are polynomial functions calculated by imposing the continuity condition on the field variable. Whereas in the case of Hermite interpolation, the polynomial functions are calculated by imposing an additional condition on the continuity of the first derivative of the field variable.

1. Lagrange Interpolation

The Lagrange interpolation of degree n , which goes through $n+1$ control points (x_j, q_j) is calculated by constructing a polynomial:

$$p(x) = p_n x^n + p_{n-1} x^{n-1} + \dots + p_1 x^1 + p_0 \quad 2.3$$

The unknown coefficients p_i , are calculated by solving $n+1$ equations for the unknowns by imposing the condition, $p(x_i) = q_i$.

Also, these functions can be calculated using a simpler formula, given by Lagrange in 1795 [34].

$$P(x) = \sum_{j=1}^n B_j(x) q_j \quad 2.4$$

$$\text{where, } B_j(x) = \prod_{\substack{i=0 \\ i \neq j}}^n \frac{(x-x_i)}{(x_j-x_i)} = \frac{(x-x_0)}{(x_j-x_0)} \frac{(x-x_1)}{(x_j-x_1)} \dots \frac{(x-x_n)}{(x_j-x_n)} \quad 2.5$$

The Lagrange interpolation functions have some unique properties as discussed below:

a. A data set of n control points can be interpolated by a unique Lagrange polynomial of degree $n-1$.

b. $B_i(x_j) = \delta_{ij}$, where δ_{ij} is the kronecker delta, and is defined by:

$$\delta_{ij} = \begin{cases} 0 & \text{for } i = j \\ 1 & \text{for } i \neq j \end{cases} \quad 2.6$$

$$\text{c. } \sum_{j=1}^n B_j(x) = 1 \quad 2.7$$

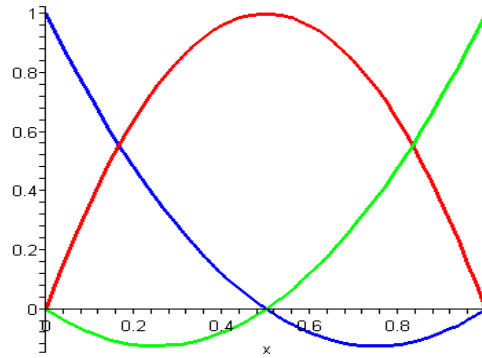


Fig.2 Quadratic Lagrange interpolation functions.

Figure 2 shows quadratic Lagrange interpolation functions for $0 < x < 1$, with a set of three functions required to interpolate a set of exactly three control points. As can be seen from the figure, only one function has a value 1 at each node, all other are zero. Figure 3 shows the interpolated curve obtained using the quadratic Lagrange functions for the data set $[[0, 0], [1/2, 1], [1, 4]]$. As can be seen, the interpolated curve passes through all the three points. But, if the number of control points are increased, the order of the polynomial needs to be increased, which can result in computational

complications. There is another option for interpolating more number of control points, i.e. using lower degree polynomials for a sub-set of the entire control point set. In conventional FEM, the Lagrange interpolation functions are used in a similar way, i.e. by dividing the control points into sub-sets called elements.

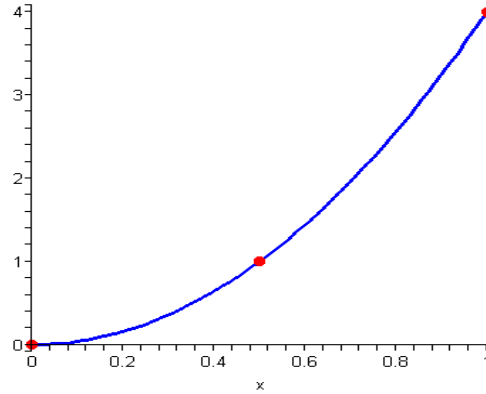


Fig.3 Curve obtained by interpolating a set of three points using quadratic Lagrange shape functions.

Also, the order of continuity of the curve obtained using Lagrange interpolation is C^0 . The C^0 continuity implies that only the continuity of the field variable is maintained between elements. To attain a higher order continuity for particular problems, Hermite interpolation is used. In the case of Hermite interpolation, the continuity of the field variable and its first derivative is maintained. i.e the order of continuity for the curve obtained using Hermite interpolation is C^1 .

The Lagrange and Hermite interpolation are widely used in the finite element applications due to their ease of use. Although, these interpolants can provide sufficient accuracy, they are computationally costly. Also for a higher accuracy, and smoothness, higher degree polynomial functions or the use of piecewise defined lower degree polynomials are required, which makes the program less efficient. The higher order Lagrange functions also exhibit oscillatory behavior in case of a higher degree polynomial. With an increase in the number of nodes, the order of the polynomial has to

be increased, hence resulting in the unwanted oscillatory behavior. i.e., although the polynomial matches the control points, in between the control points the polynomial can vary largely.

For a large number of nodes, the oscillation problem can be taken care of by using piecewise polynomial functions. The use of piecewise polynomials allows the interpolation of the same number of control points using a collection of lower degree polynomials. Piecewise Lagrange and Hermite interpolation are used in traditional FEM. The analysis domain is partitioned into regions called elements, and the polynomial functions are defined over each element.

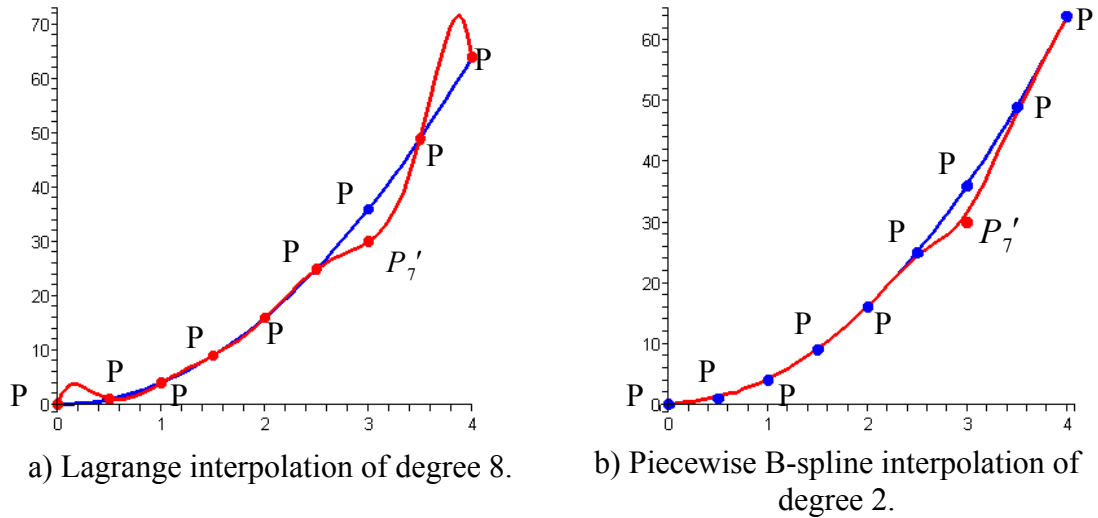


Fig.4 The effect of change in one coordinate in the interpolation of a set of data points with the change in interpolation type.

Figure 4 show the fitted curves for two data sets of 9 control points using two different types of interpolation functions. Figure 4(a) is obtained by fitting the set of 9 control points using Lagrange interpolation of degree 8. The same figure also shows the change in the fitted curve when a control point P_7 is changed to P'_7 . A similar plot is obtained as shown in fig. 4 (b), showing the curve fit for the same data points using

piecewise cubic B-spline interpolation functions. For fitting a set of 9 data points there is a unique representation possible using a Lagrange polynomial of degree 8. In case of piecewise the same set can be fitted using polynomials of degree 2. With the change in just one control point P_7 , in the case of Lagrange interpolation, a kink is seen between P_1 and P_2 , whereas in the case of piecewise, only the curve between P_6 and P_8 is altered. This is due to the local control property of the piecewise definition of interpolation. Also, in the case of Lagrange, there is an oscillation of the curve that is observed between P_6 and P_9 .

Even though the piecewise Lagrange interpolation would also have fit a large number of control points, it does not have a continuity of derivatives between the pieces. A B-spline interpolation function takes care of the continuity problem. B-spline curves are also piecewise defined curves, with each component of the curve of degree d . These piecewise defined functions allow fitting a large number of control points, and also maintain a C^{d-1} continuity, where d is the degree of the piecewise defined functions. The definition and the properties of the B-Spline functions are discussed in the following section.

2. B-spline Interpolation

A B-spline curve is a piecewise defined polynomial function connected continuously by different curve segments. The piecewise definition allows approximation of a large number of control points using lower order polynomials. In the case of B-spline interpolation, the number of control points (n) are independent of the order of the polynomials used (k). The order of a B-spline curve is one more than the degree d of the polynomial used to represent the curve. For example, for a cubic polynomial the degree is 3 and the order of the B-spline curve is 4. Cubic spline functions are extensively used in CAD/CAM to define surfaces and curves.

Any curve can be represented as a parametric curve, i.e the coordinate x is represented as a function of a parameter t . A parametric B-spline function can be defined by:

$$P(t) = \sum_{i=0}^n P_i B_{i,k}(t) \quad \text{where } t_0 < t < t_m \quad 2.8$$

where P_i are $n+1$ control points, and $B_{i,k}$ are the $n+1$ basis functions of order k associated with the control points. If the order of the polynomial used is equal to the number of control points, then the definition of the basis functions no longer remains piecewise. Hence, for a piecewise definition of the basis functions k should be less than $n+1$. For example, if a 7th order polynomial is used for fitting 8 control points, there would be only one function definition for the entire curve. For the same number of control points, if cubic B-Spline interpolation is used (order 4), the basis functions would have a piecewise definition with 5 different pieces.

The intervals in which the spline functions are defined in the parameter space (t) are called knot intervals. There is a single function definition between two different knot values. The vector containing the knot points is called a knot vector, and is formed by arrangement of the knots in an increasing sequence, i.e., knot vector can be written as:

$$T = [t_0, t_1, t_2, \dots, t_{m-1}, t_m], \text{ where } t_i \leq t_{i+1}$$

Figure 5 shows a piecewise defined cubic B-spline basis function composed of 4 different function definitions over 4 knot intervals. The total number of knots (m) required to interpolate n control points using polynomials of degree k can be calculated using $m = n + k + 1$. The interval between any two knots is also called the knot spacing, and it can be zero as well. The basis function $B_{i,k}$ are nonzero only in the domain $t_i < t < t_{i+k+1}$, the definition of the basis functions $B_{i,k}$ is different for every knot interval.

The knots govern the polynomials and thus the shape of the interpolated curve. A curve with equal intervals in the parametric space, i.e with equally spaced knots, is called a uniform B-spline. In the case of unequal spacing between the knots, the resulting curve is called a non-uniform B-spline curve. The effect of knot spacing on the fitted curve will be discussed in more detail in the further sections.

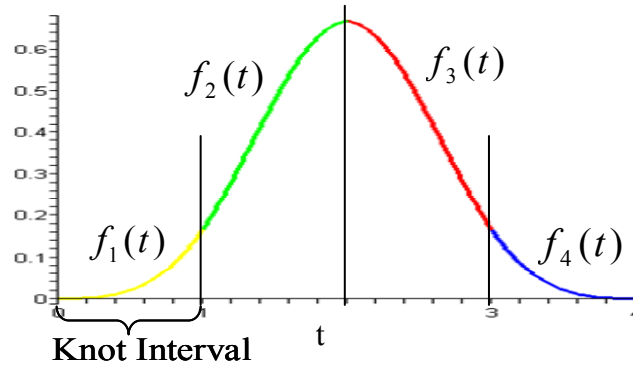


Fig.5 Piecewise defined cubic B-Spline basis function.

The basis functions defined for the entire domain are non-zero for sub-domains only, hence they influence a limited domain in a curve. This property is identified as the local support property. This enables low influence or effect of change in one control point on a far off control point. i.e when a control point is modified, the curve would be altered only in the domain where the associated basis function is non-zero. The resulting curve possesses C^{n-1} continuity at all points.

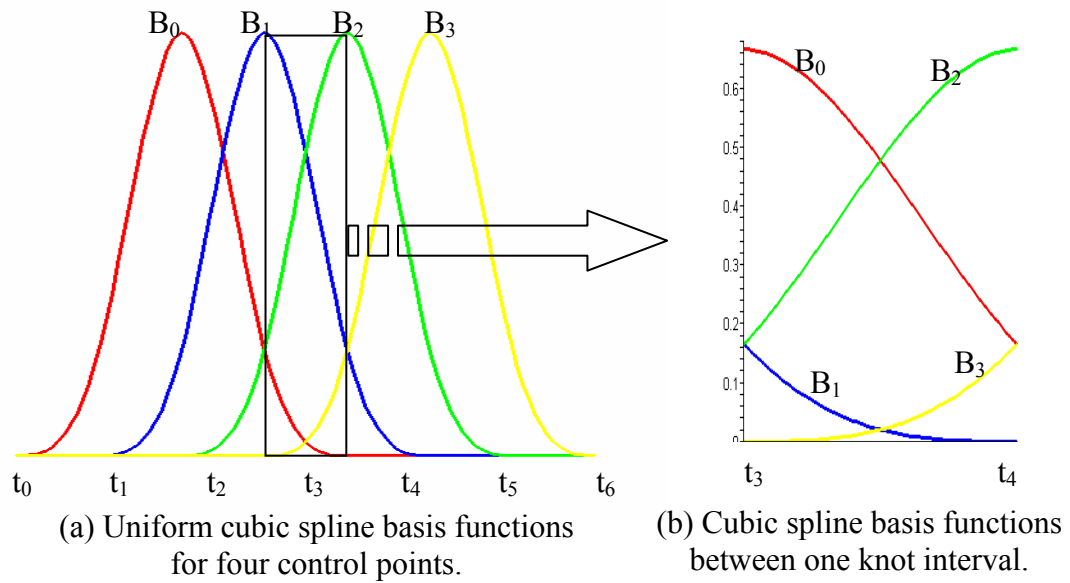


Fig.6 Uniform cubic basis functions.

As can be seen in fig. 6, basis function B_0 is non- zero between knots t_0 and t_3 only, hence it has a local support between 4 knot intervals. Figure 6(b) shows cubic spline basis functions between the knot values t_3 and t_4 . It can be noted that the functions defined between all the knot intervals are identical, except that they are offset. It can also be noticed that there is more than 1 non-zero function at a particular knot value, as compared to Lagrange interpolation functions where only one function has a value 1, and all other are zero.

3. Summary of Properties [3] of B-spline Basis Functions:

a. The basis functions are positive.

$$\text{i.e } B_{i,k}(t) \geq 0 \text{ for all } t$$

b. Local support

$$\text{i.e } B_{i,k}(t) \neq 0 \text{ for } t_i < t < t_{i+k+1}.$$

The local support property for linear and quadratic spline functions can be seen in fig.7. The support interval for $B_{1,1}$ is between knots t_1 and t_3 . Similarly, for a quadratic basis function $B_{3,2}$, the support interval is between knots t_3 and t_6 .

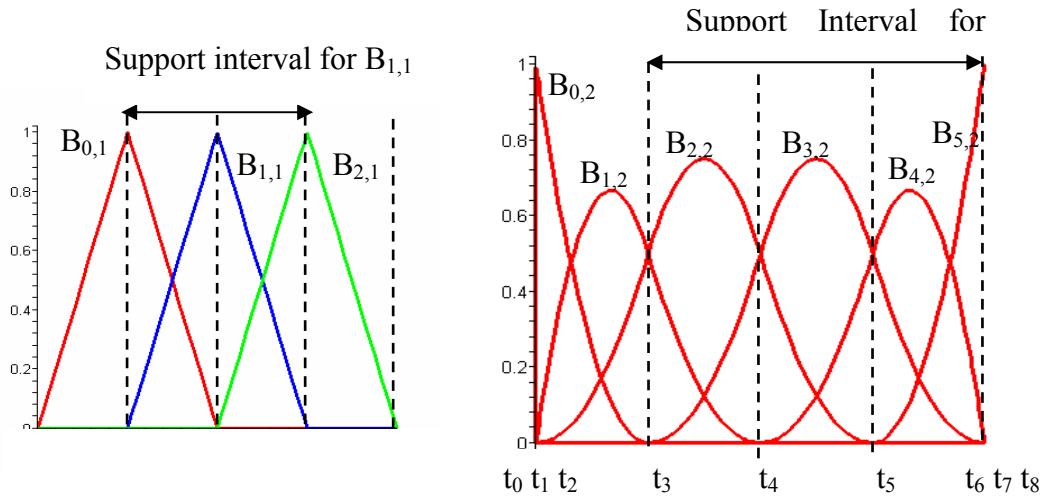


Fig. 7 Local support interval for linear and quadratic spline functions.

c. The sum of all basis functions at a particular value of t equals unity.

$$\text{i.e.} \quad \sum_{i=0}^n B_{i,k}(t) = 1$$

d. B-spline functions are recursive in nature. i.e the B-spline functions at a higher order can be recursively calculated using the lower order splines.

The B-spline functions for the first order are given by:

$$B_{i,0}(t) = \begin{cases} = 1, & t \in [t_i, t_{i+1}) \\ = 0, & \text{otherwise} \end{cases}$$

The higher order B-Spline functions can be recursively calculated using :

$$B_{i,k}(t) = \frac{t - t_i}{t_{i+k} - t_i} B_{i,k-1}(t) + \frac{t_{i+k+1} - t}{t_{i+k+1} - t_{i+1}} B_{i+1,k-1}(t)$$

e. k is the order of the polynomials used to define the basis functions.

The polynomials have to be a minimum of linear order (i.e. $k \geq 2$), and $k \leq n + 1$.

f. The basis functions have exactly one maxima.

4. Knot Vectors

The shape of the basis functions is largely governed by the knot spacing in a knot vector. As discussed earlier, the number of knots is given by $m = n + k + 1$, where n are the number of control points, and k is the order of the B-spline curve. The knot vector consists of ascending knot values, which can be repeating as well. e.g. $[0, 0, 1, 2, 3, 4, 5, 5]$ can be used to fit four control points using cubic basis functions. The difference between two corresponding knot values is called the knot spacing. For a repeating set of knot values, the knot spacing can be zero as well. The shape of the basis functions is dependent on the knot spacing rather than the actual knot value. For example, the two figures in fig. 8 show the basis functions for two different knot vectors $[0, 1, 2, 3, 4, 5, 6, 7]$, and $[0, 2, 4, 6, 8, 10, 12, 14]$. Even though the actual values in the knot vector are different, the knot spacing is uniform in both the cases, hence the basis functions have the same shape.

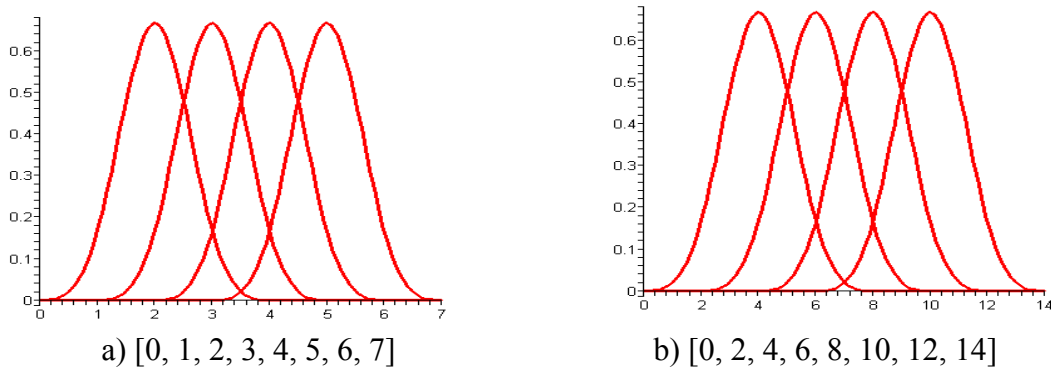


Fig.8 Effect of knot spacing on shape of basis functions.

Knot Vectors can be generally classified as uniform, open uniform and non-uniform.

Uniform Knot Vectors

In case of a uniform knot vector the knot spacing is a constant.

i.e $t_{i+1} - t_i = \text{constant}$

e.g. [0, 1, 2, 3, 4, 5, 6, 7, 8, 9] or [0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8]

In the case of a uniform knot vector, the basis functions are periodic uniform. i.e between each knot interval, the basis functions are identical except that they are translated along the knot axis. As can be seen from fig. 9, with a uniform knot vector, we get periodic B-splines. Figure 9(a) shows the periodic B-splines of cubic order, with a uniform knot vector [0,1,2,3,4,5,6,7], and fig. 9 (b) shows the periodic B-splines of linear order, with a uniform vector [0,1,2,3,4], as can be seen from the two figures, the basis functions shown in different colors are identical in each figure.

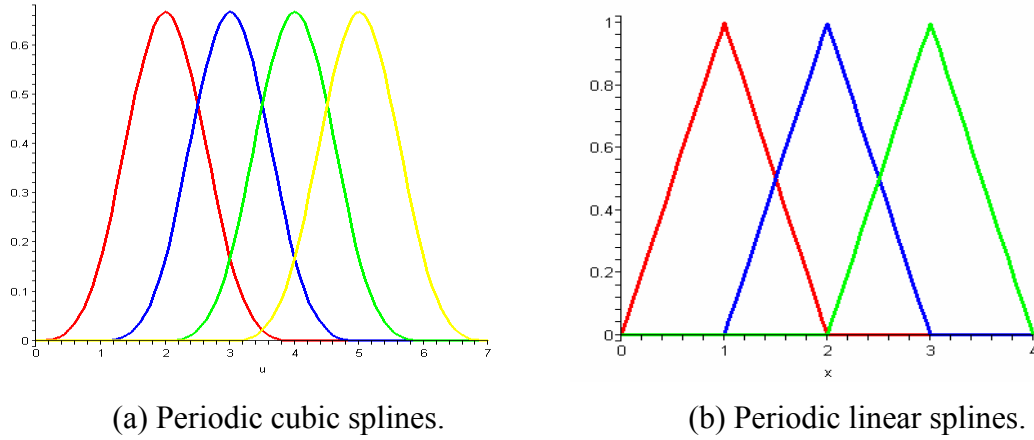


Fig. 9 Periodic splines using uniform knot vectors.

Open Uniform Knot Vectors

An open uniform knot vector has k repeating knots at the two ends, and equal knot spacing between the inside knots. The open uniform knot vector can be written as:

$$\begin{aligned}
 t_i &= t_0 & i &\leq k \\
 t_{i+1} - t_i &= \text{constant} & k &\leq i < n+1 \\
 t_i &= t_{n+k} & i &\geq n+1
 \end{aligned}$$

e.g for a set of 6 control points and basis functions of order 4, the open uniform knot vector can be written as: $T = [0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4]$

The repetition of knots in a knot vector is known as multiplicity. It can be measured by the number of times a knot is repeated in a knot vector. For example, given a knot vector $[0, 0, 0, 1, 4, 6, 6]$, the multiplicity of the knot 0 is 3, and the multiplicity of knot 6 is 2. The repetition of knots in a knot vector, brings the curve closer to the control point to be interpolated. The multiplicity also reduces the continuity of the curve near the associated control point by the multiplicity value. i.e for basis function of degree p , the continuity of the curve is $C^{p-\text{multiplicity}}$. If the multiplicity is equal to the order of the basis functions, the curve definitely passes through the associated control point. When the multiplicity is equal to the order of the basis functions, only one basis function has a value equal to 1, and all other equal zero. As can be seen from fig.10, the knot values at

the ends have multiplicity equal to the order of the curve, i.e. 3. Only one basis function at the ends has a value 1, all other basis functions are zero.

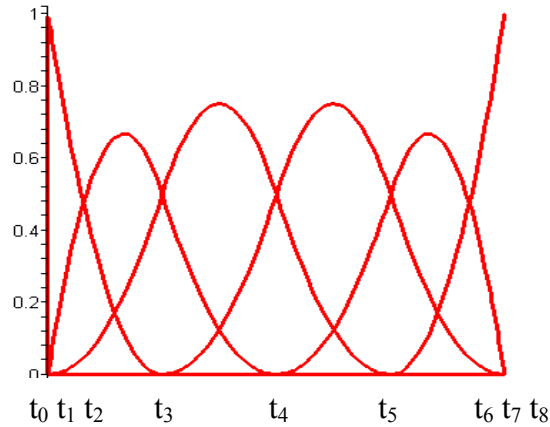


Fig.10 Open uniform quadratic basis functions for knot vector $[0,0,0,1,2,3,4,4,4]$.

Figure 11 shows four basis functions for open uniform knot vector $[0, 0, 0, 0, 1, 1, 1, 1]$. The number of times the end knots are repeated are equivalent to the order of the B-spline curve, 4 in this case. The basis functions in this case are same as the cubic Bezier polynomials [3]. The fig.10 shows quadratic basis functions for an open uniform knot vector $[0,0,0,1,2,3,4,4,4]$. As can be seen, at the ends, only one basis function has a value 1, all other are zero, forcing the curve to pass through the associated control point.

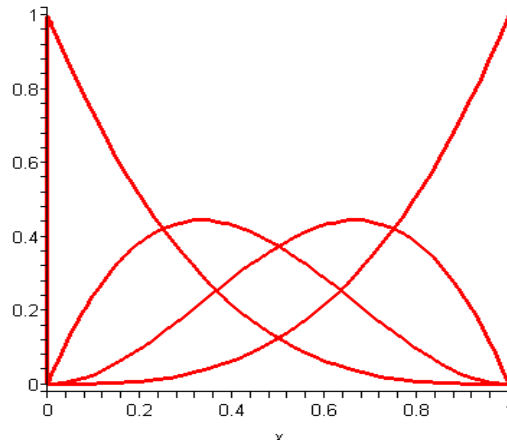


Fig.11 Open uniform cubic basis functions for knot vector $[0,0,0,0,1,1,1,1]$.

Non- Uniform Knot Vectors

This is any set of ascending numbers used to represent knot values, e.g. $[0, 0.3, .5, 0.7, 0.7, 0.75, 0.78, 3]$. The quadratic basis functions for a non-uniform knot vector $[0, 0, 0, 0.5, 1, 1.5, 2, 2.5, 2.5]$, can be seen in fig.12. Since the knots on the left end have been repeated three times, one basis function has a value 1 at that end and all other basis functions are zero.

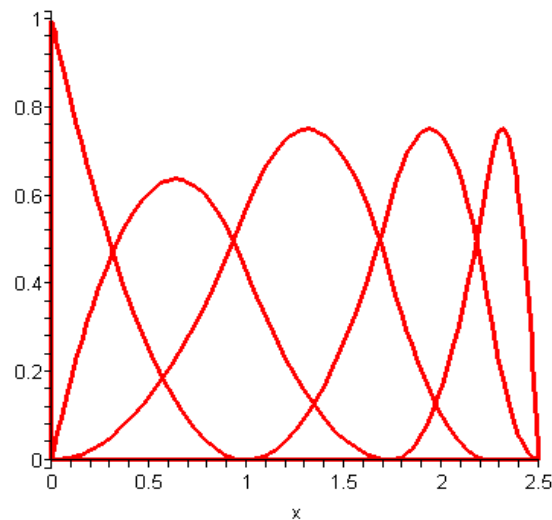


Fig.12 Quadratic basis functions for a non-uniform knot vector $[0, 0, 0, 0.5, 1, 1.5, 2, 2.5, 2.5]$.

To study the effect of different knot vectors on the interpolated curve, an example using control points $[[0, 0], [4, 9], [6, 10], [8, 5]]$ has been plotted in fig.13. The first curve in the figure represents the curve obtained by using a periodic set of basis functions using a uniform knot vector. The second one represents a curve obtained by an open uniform knot vector, and the third one using a non-uniform knot vector. As can be seen from the figure, for the same set of control points, and same order of basis functions, but different knot vectors, the interpolated curves are different. In the case of a uniform knot vector, the curve doesn't pass through the two end control points, P_0 and P_n . The repetition of knots at the ends in the open-uniform knot vector forces the curve to pass through the end control points, as can be seen in fig. 13(b). In the third case, since the knots are only repeated at the left end, the curve only passes through the left end control point.

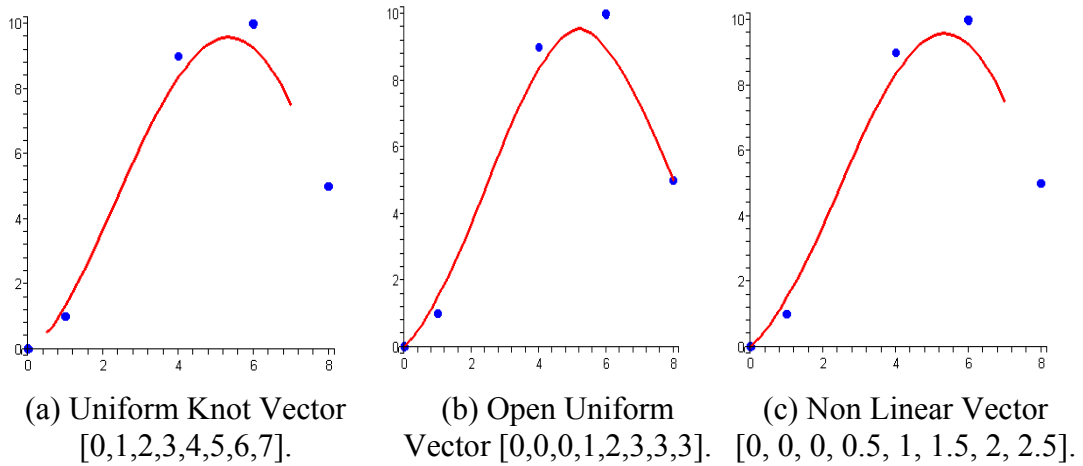


Fig.13 Comparison of B-spline curves for control points $[[0, 0], [4, 9], [6, 10], [8, 5]]$ using three different knot vectors.

5. B-Spline for Two Dimensions

The interpolation functions for two dimensions are calculated by the tensor product of the B-spline interpolation functions in the t_1 and t_2 directions. Figure 14 shows a two dimensional region for which interpolation functions need to be calculated.

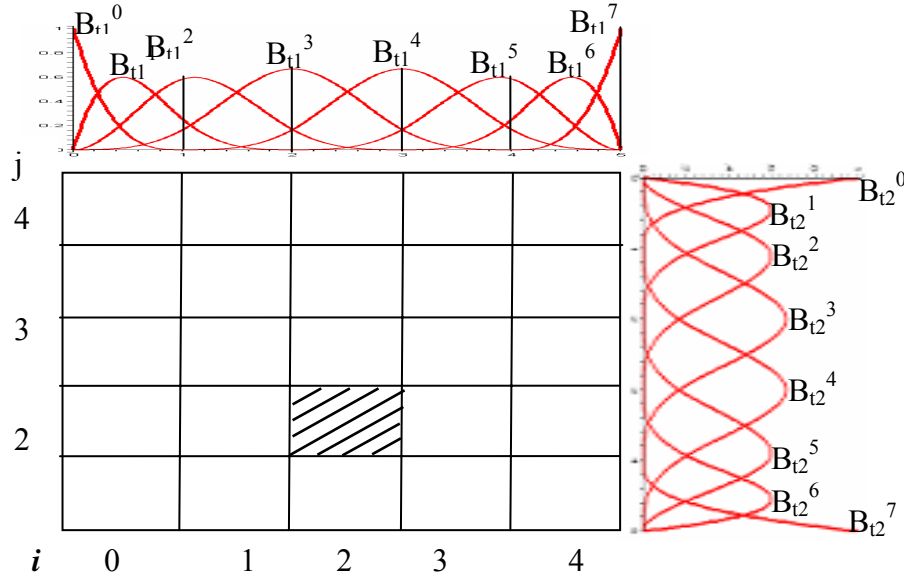


Fig. 14 B-spline patches for two dimensions.

The interpolation functions associated with each two dimensional sub-region are calculated on regions defined by knot spaces in each dimension. Each of these sub-regions, called *patches*, has 16 interpolation functions. The 16 functions are calculated by tensor product of 4 functions from t_1 with the 4 functions in t_2 direction. The functions required to calculate the interpolation functions for each patch can be calculated by the knot intervals in the t_1 and t_1 directions associated with it. For example, for the shaded patch as shown in the fig. 14, the knot interval in t_1 is 2 and in the t_2 direction is 3. The interpolation functions are calculated by taking the tensor product of $[B_{t1}^2, B_{t1}^3, B_{t1}^4, B_{t1}^5]$ and $[B_{t2}^3, B_{t2}^4, B_{t2}^5, B_{t2}^6]$. The tensor product of a patch with i^{th} knot interval in t_1 and j^{th} in t_2 can be written as:

$$\begin{bmatrix} B_{t_1}^i \\ B_{t_1}^{i+1} \\ B_{t_1}^{i+2} \\ B_{t_1}^{i+3} \end{bmatrix} [B_{t_2}^j \ B_{t_2}^{j+1} \ B_{t_2}^{j+2} \ B_{t_2}^{j+3}] = \begin{bmatrix} B_{t_1}^i B_{t_2}^j & B_{t_1}^i B_{t_2}^{j+1} & B_{t_1}^i B_{t_2}^{j+2} & B_{t_1}^i B_{t_2}^{j+3} \\ B_{t_1}^{i+1} B_{t_2}^j & B_{t_1}^{i+1} B_{t_2}^{j+1} & B_{t_1}^{i+1} B_{t_2}^{j+2} & B_{t_1}^{i+1} B_{t_2}^{j+3} \\ B_{t_1}^{i+2} B_{t_2}^j & B_{t_1}^{i+2} B_{t_2}^{j+1} & B_{t_1}^{i+2} B_{t_2}^{j+2} & B_{t_1}^{i+2} B_{t_2}^{j+3} \\ B_{t_1}^{i+3} B_{t_2}^j & B_{t_1}^{i+3} B_{t_2}^{j+1} & B_{t_1}^{i+3} B_{t_2}^{j+2} & B_{t_1}^{i+3} B_{t_2}^{j+3} \end{bmatrix} \\
= \begin{bmatrix} B_0 & B_1 & B_2 & B_3 \\ B_4 & B_5 & B_6 & B_7 \\ B_8 & B_9 & B_{10} & B_{11} \\ B_{12} & B_{13} & B_{14} & B_{15} \end{bmatrix}$$

The interpolation functions obtained after the tensor product are renumbered so as to obtain a one dimensional array. The numbering of the interpolation functions for each patch is as shown in the equation above. Each patch has 16 interpolation functions associated with it which are shared by the neighboring patches as in the case of one-dimensional interpolation functions.

The following chapters discuss the implementation of cubic B-spline functions to one dimension and two-dimension finite element method.

CHAPTER III

B-SPLINE FINITE ELEMENTS- 1D

In a conventional finite element model, a complex region is divided into simpler finite regions called elements. The boundaries of these elements are placed according to the discontinuities of the geometry or specification of the boundary conditions. The displacement field in each element is approximated by interpolation functions multiplied by the displacement at some particular points in the element called nodes. The interpolation functions and number of degrees of freedom for a specific element are of either Lagrange or Hermite type of a particular order. For each of these elements, the number of equations is the same as the number of degrees of freedom. These equations are then assembled with the help of coincident nodes for the neighboring elements. The boundary conditions are then applied to the equations to solve for the unknown displacements at the nodes.

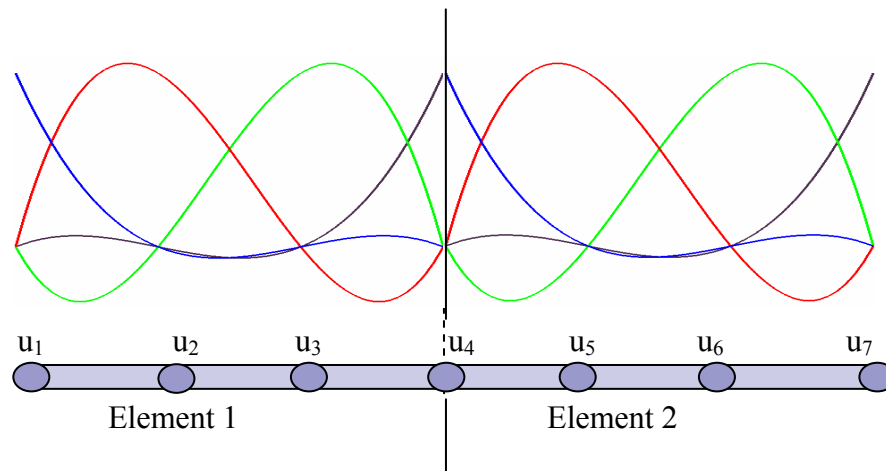


Fig.15 Two cubic Lagrange elements.

Figure 15 shows two cubic Lagrange elements in case of regular finite elements, with four degrees of freedom per element. As can be seen in the fig, there is no overlap in the function definitions between the two elements. Hence, the degrees of freedom associated in one element are not shared by the adjoining elements except at the boundary of the adjacent elements. To calculate the element matrices, there is a need to evaluate integrals. Due to algebraic complexity, it is always not possible to calculate the integrals exactly, hence numerical integration is used to calculate the integrals. The most common method used is the Gauss- Legendre method in which the integral can be calculated for an interval $[-1, 1]$. To evaluate the integral using this method, the physical coordinates of the problem(x) need to be transformed to the local coordinate system (ξ). This local coordinate is also called the normalized coordinate system, and has values ranging from -1 to 1. The interpolation functions shown in fig.15 associated with each element are defined in this local coordinate system in the range $[-1, 1]$.

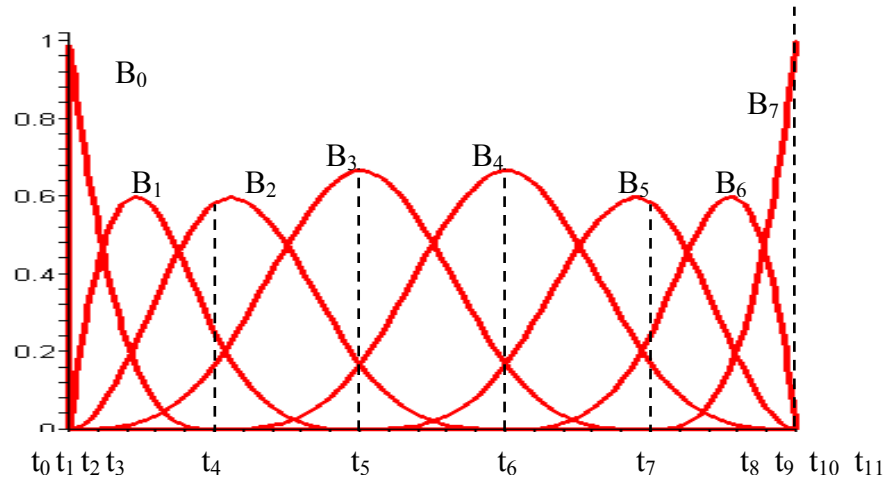


Fig.16. Open uniform B-spline basis functions of cubic order defined for each cluster.

In the implementation of B-spline finite element, the approach for solving for the unknowns is similar, except for the type of interpolation functions and the associated implementation details. In this case the geometry is divided into simpler regions called clusters, according to the discontinuities similar to subdivision into elements in the

regular finite element method. The solution for this cluster is approximated by cubic B-spline functions. An open uniform knot vector, as shown in fig. 16 is used to define the basis functions for each cluster. The figure shows 8 basis functions, which are associated with 8 corresponding control points. The order of the B-spline curves is four, hence for convenience, the basis functions are written as B_i rather than $B_{i,4}$. The number of control points and corresponding basis functions can be varied according to the problem needs. The open uniform knot vector ensures that the solution matches the value at the boundary nodes.

In conventional finite elements, each of the elements represents the entire integration domain. The coordinates of the boundary nodes of each element are transformed into the normalized coordinates $(-1, 1)$ in the case of regular finite elements. But in the case of B-spline finite elements, the knot intervals are the integration domain, and hence the coordinates of each knot interval are transformed to local coordinates. As can be seen in fig.17, each interval between non-repeating knots is considered as an integration domain.

Since cubic spline functions have been used in this implementation, the number of basis functions associated with each knot interval is 4. Even when the total number of basis functions associated with the entire cluster are 8, as shown in fig. 17, only the non-zero basis functions in that interval are integrated. Since only four functions are used in the integral in each knot interval, these integral domains take advantage of the local support property of the B-spline basis functions, and avoid the integration of zeroes for a greater numerical efficiency. Figure 17 shows basis functions associated with each knot interval, and the corresponding associated degrees of freedom. A stiffness matrix for each knot interval is calculated, using the regular finite element formula for stiffness matrix. These stiffness matrices are then assembled to form the stiffness matrix of the cluster, using the shared degrees of freedom amongst the knot intervals. For example, in every sub-cluster, the solution is expressed by the four interpolation functions and the associated control points, out of which three are shared with the intervals on the left and

three with the intervals on the right. This property is distinctly different from the ordinary finite element method.

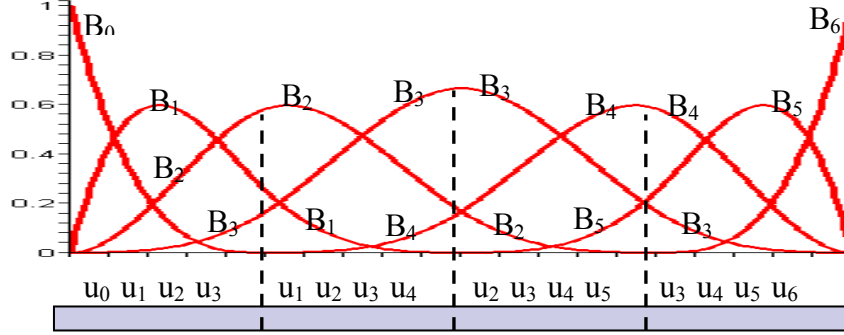


Fig. 17 Integration domains for B-spline finite elements.

For n control points, and k the order of the polynomials, the number of knots ' m ' is given by ' $n + k + 1$ '. Since there is a repetition of ' k ' knots at both ends in the case of an open uniform knot vector, the number of knot intervals is calculated to be ' $n - k + 1$ '. Since, only cubic order basis functions have been considered herein, the number of integral domains = $n - 3$. Each of these knot intervals is defined by four basis functions ($B_i, B_{i+1}, B_{i+2}, B_{i+3}$) and corresponding control points ($u_i, u_{i+1}, u_{i+2}, u_{i+3}$), where i is the knot interval number. These independent integral domains have overlapping control points and basis functions, as can be seen in fig. 17.

The number of knot intervals for each cluster can be specified by the user according to the problem. The basis functions for each knot interval are defined separately, as at any one given time only the basis functions associated with a single knot interval are required for integration. The number of knot intervals can be increased as shown in fig. 18, by adding the repeated set of basis functions in between the left and right boundary knot intervals. The basis functions at the left boundary for all cases are the same and are thus labeled as L-1, and L-2. Similarly the basis functions at the right boundary for all cases, labeled as R-1 and R-2 are similar, they are just translated on to the right depending on the number of knot intervals in the cluster. The repeated set is inserted and then translated according to the number of knot intervals in each cluster. In

the case of 1 knot interval only for a cluster, the number of basis functions for the entire cluster is 4, and are shown in fig. 19. The case of 2, 3 and 4 knot intervals do not follow the same pattern, and thus have to be defined separately. For the case of 1 knot interval, the functions are not chosen out of the five set of functions, and are rather stored separately as a special case. The case of 2 and 3 knot intervals have not been implemented, but can be defined similarly.

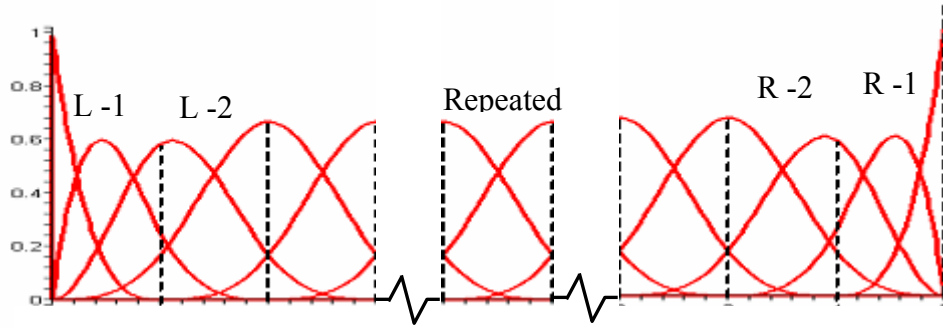


Fig. 18 B-spline basis functions for different knot intervals.

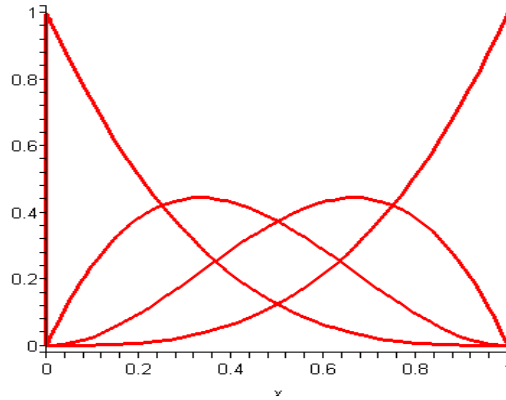


Fig.19 Basis functions for a cluster with single patch.

The boundary and the internal set of functions are programmed in a set of five sub-routines in the code. The function derivatives are also calculated and programmed in a set of five more sub-routines. Since the integration for each knot interval is calculated independent of other intervals, there are only four basis functions required inside the

integration loop. These basis functions are those that are associated with the knot interval over which the integral is calculated.

To calculate the stiffness matrix for a knot interval, and subsequently the cluster, the material properties are required. A separate function is written to read the modulus as in the case of regular finite elements. There is no difference in the declaration or use of this function. Besides the specification of the mesh properties, the material properties, and the element properties are specified similar to the case in regular finite elements. The properties input by the user are used in the program to calculate the stiffness matrix. The stiffness matrix for each of the knot intervals is calculated using the associated basis functions and their derivatives. In the case of regular finite element analysis, each element is normalized to local coordinates, for implementing numerical integration. In this case, each knot interval, is normalized to local coordinates (-1, 1). Hence, the process of calculation of the stiffness matrix for each knot interval involves two transformations, which will be discussed later.

In the following section the finite element development of an elastic rod using cubic B-spline interpolation functions has been discussed.

1. Elastic Rod

The differential equation for an elastic rod is given by:

$$\frac{d}{dx} \left(EA \frac{du}{dx} \right) + f = 0 \quad 3.1$$

The weak form for the equation can be written as:

$$-\int_{x_1}^{x_2} EA \frac{du}{dx} \delta \left(\frac{du}{dx} \right) dx + \int_{x_1}^{x_2} f \delta u dx + EA \frac{du}{dx} \delta u \Big|_{x_1}^{x_2} = 0 \quad 3.2$$

The displacement, u for each cluster is approximated using cubic B-spline functions B_i :

$$\begin{aligned} u &= \sum u_i B_i \quad \delta u = \sum \delta u_i B_i \\ \frac{du}{dx} &= u_i \frac{dB_i}{dx} \text{ and } \delta \frac{du}{dx} = \delta u_i \frac{dB_i}{dx} \end{aligned} \quad 3.3$$

The weak form after approximation becomes:

$$\sum \delta u_j \left[-\int_{x_1}^{x_2} EA u_i \frac{dB_i}{dx} \frac{dB_j}{dx} dx + \int_{x_1}^{x_2} f B_j dx + EA \frac{du}{dx} B_j \right] = 0 \quad 3.4$$

This can be expressed as : $[K]\{u\} = \{F\}$

The stiffness matrix, K , can be written as:

$$K_{ij} = \int_{x_1}^{x_2} EA \frac{dB_i}{dx} \frac{dB_j}{dx} dx \quad 3.5$$

The calculation of the stiffness matrix requires the global derivatives of the basis functions. The stiffness matrix for each knot interval is calculated separately and then assembled into the stiffness matrix for the cluster.

As shown in fig. 20, the cluster is transformed first into the knot coordinate system (t), which ranges from 0 to $n-3$. This is then further subdivided into knot intervals, with equal spacing. Each of the knot interval coordinates are then normalized to local coordinates.

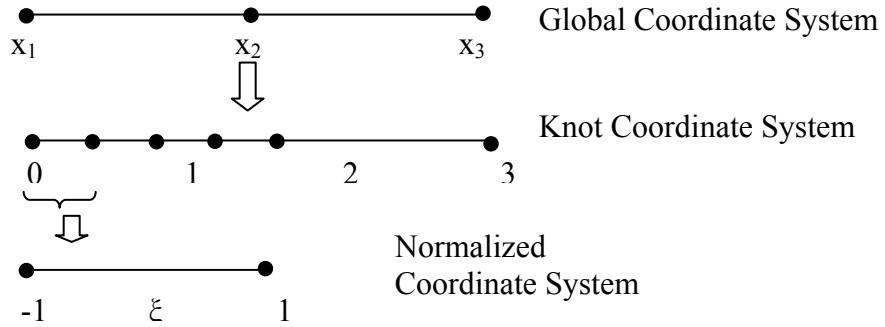


Fig.20 Transformation to normalized coordinates.

The basis functions are defined in the knot coordinate system, and to calculate the global derivatives, the jacobian is calculated for transformation from knot coordinate to the global coordinate system. Here, the global coordinate system is transformed to the knot coordinate system using Lagrange interpolation functions.

$$\text{i.e } \frac{dB_i}{dx} = \frac{dB_i(t)}{dt} \left(\frac{dx}{dt} \right)^{-1} \quad 3.6$$

$$\begin{aligned} x &= x_1 N_1(t) + x_2 N_2(t) \\ \text{where, } \frac{dx}{dt} &= x_1 \frac{dN_1(t)}{dt} + x_2 \frac{dN_2(t)}{dt} \end{aligned} \quad 3.7$$

x_1 and x_2 are the coordinates of the cluster end points as shown in the fig. 20. The Lagrange interpolation functions N_i are calculated using the values in the knot coordinate system ranging from 0 to $n-3$. The Jacobian is termed as J_1 and is written as:

$$\begin{aligned} J_1 &= \frac{dx}{dt} \\ \Rightarrow \frac{dt}{dx} &= J_1^{-1} \end{aligned} \quad 3.8$$

After transformation into the knot coordinate system, the stiffness matrix becomes:

$$\begin{aligned} K_{ij} &= \int_{t_1}^{t_2} EA \frac{dB_i}{dt} \left(\frac{dt}{dx} \right) \frac{dB_j}{dt} \left(\frac{dt}{dx} \right) \left(\frac{dx}{dt} \right) dt \\ &= \int_{t_1}^{t_2} EA \frac{dB_i}{dt} (J_1^{-1}) \frac{dB_j}{dt} (J_1^{-1}) (J_1) dt \end{aligned} \quad 3.9$$

The stiffness matrix and the force vector integral are calculated using numerical integration. The Gauss-Legendre method calculates the integral for a range $[-1, 1]$, hence the integrand is normalized into $[-1, 1]$ coordinates. For the transformation, the standard Lagrange shape functions used in conventional finite element are used.

$$t = \sum_{i=0}^1 t_i S_i(\xi) \quad 3.10$$

$\frac{dB_i}{dt}$ is a function of t , where t is replaced by the expression $t = \sum_{i=0}^1 t_i S_i(\xi)$ to get a function of ξ .

$$J_2 = \frac{dt}{d\xi} \quad 3.11$$

Hence the expression for the stiffness matrix for each knot interval becomes,

$$\begin{aligned}
K_{ij} &= \int_{-1}^1 EA f_i(\xi) (J_1^{-1}) f_j(\xi) (J_1^{-1}) (J_1) \frac{dt}{d\xi} d\xi \\
&= \int_{-1}^1 EA f_i(\xi) (J_1^{-1}) f_j(\xi) (J_1^{-1}) (J_1) (J_2) d\xi
\end{aligned}
\tag{3.12}$$

The stiffness matrix terms are calculated for each knot interval separately using numerical integration of the expression. The equations for these knot intervals are then assembled using the overlapping control points, to obtain the stiffness matrix for the entire cluster. The stiffness matrix for the clusters, are then further assembled into the global stiffness matrix, like regular finite elements. This matrix is then assembled using the degree of freedom connectivity for each knot interval to find out the stiffness matrix for each cluster. In the case of 1D, the connectivity for the sub-clusters is easy and just follows a pattern of sharing 3 degrees of freedom in between every knot interval. i.e, if u_0, u_1, u_2, u_3 are associated with the first interval, then u_1, u_2, u_3, u_4 are associated with the next knot interval. Using this pattern, the stiffness matrix (K) for the cluster is calculated. The assembly pattern for the stiffness matrix for a cluster, is shown in fig.21.

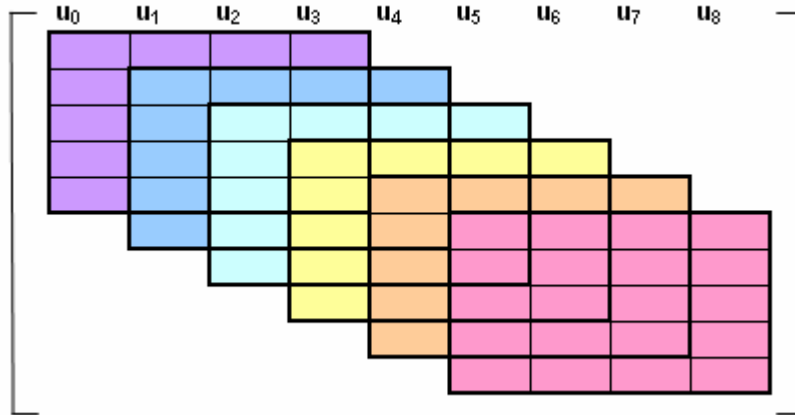


Fig.21. Stiffness matrix assembly for a cluster for 1D analysis.

CHAPTER IV

B-SPLINE FINITE ELEMENTS- 2D

An earlier section discussed the finite element implementation of a bar element with a single variable using B-spline interpolation. In this section, the B-spline basis functions used as interpolation functions for plane elasticity problems are discussed. In this case there are two dependent variables, u and v , which are the displacements in two different directions x and y .

1. Plane Elasticity

A three dimensional elasticity problem can be solved using the plane elasticity conditions if there is no variation in the third direction with respect to the third dimension (z). In that case, the governing equations are without the derivatives in the z direction. The plane elasticity problems can be categorized into two main categories: *plane stress* and *plane strain*.

Plane stress conditions are applied to problems where the thickness in the third direction is small as compared to the in-plane dimensions. In this case the stresses in the z direction are considered to be zero.

$$\text{i.e} \quad \sigma_{xz} = \sigma_{yz} = \sigma_{zz} = 0 \quad 4.1$$

Plane strain conditions are generally applied where the thickness in the z -direction is large as compared to the in-plane dimensions. It is solved based on assumption that there are no displacements in the z direction.

$$\text{i.e} \quad \varepsilon_{xz} = \varepsilon_{yz} = \varepsilon_{zz} = 0 \quad 4.2$$

Kinematics

For a general plane elastic condition, the strain displacement relations for the plane elasticity condition are given by:

$$\varepsilon_x = \frac{\partial u}{\partial x} \quad \varepsilon_y = \frac{\partial v}{\partial y} \quad \varepsilon_{xy} = \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \quad 4.3$$

where, u and v are the displacement variables in the x and y direction, respectively and ε_{xy} is the engineering shear strain.

Constitutive Law

The constitutive relationship for the plane- elasticity case is given by:

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \sigma_{xy} \end{Bmatrix} = \begin{bmatrix} C_{11} & C_{12} & 0 \\ C_{12} & C_{22} & 0 \\ 0 & 0 & C_{33} \end{bmatrix} \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_{xy} \end{Bmatrix} \quad 4.4$$

where C_{ij} ($i=1..3, j=1..3$) are the material constants and can be calculated from the engineering constants according to the plane- stress or plane strain condition as follows.

The material constants for plane stress are

$$\begin{aligned} C_{11} &= \frac{E_1}{1-\nu_{12}\nu_{21}} & C_{22} &= \frac{E_2}{1-\nu_{21}\nu_{12}} \\ C_{12} &= \nu_{21}C_{11} & C_{33} &= G_{12} \quad \frac{\nu_{12}}{\nu_{21}} = \frac{E_1}{E_2} \end{aligned} \quad 4.5$$

$E_1, E_2, \nu_{12}, \nu_{21}, G_{12}$ are the engineering constants.

The material constants for the plane strain case are

$$\begin{aligned} C_{11} &= \frac{E_1(1-\nu_{12})}{(1-\nu_{12}-2\nu_{12}\nu_{21})} & C_{22} &= \frac{E_2(1-\nu_{21}\nu_{12})}{(1+\nu_{12})(1-\nu_{12}-2\nu_{12}\nu_{21})} \\ C_{12} &= \frac{E_2\nu_{12}}{(1-\nu_{12}-2\nu_{12}\nu_{21})} & C_{33} &= G_{12} \quad \frac{\nu_{12}}{\nu_{21}} = \frac{E_1}{E_2} \end{aligned} \quad 4.6$$

Equilibrium Equations

The equilibrium equations for a plane elasticity problem with a constant thickness are

$$\begin{aligned} \frac{\partial \sigma_x}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} + f_x &= 0 \\ \frac{\partial \sigma_y}{\partial y} + \frac{\partial \sigma_{xy}}{\partial x} + f_y &= 0 \end{aligned} \quad 4.7$$

where, f_x and f_y are the body forces in the x and y directions respectively.

Boundary Conditions

The boundary conditions for plane elasticity are given by:

Natural boundary conditions:

$$\begin{aligned}\sigma_x n_x + \sigma_{xy} n_y &\equiv T_x \\ \sigma_{xy} n_x + \sigma_y n_y &\equiv T_y\end{aligned}\tag{4.8}$$

where n_x, n_y are the components of the unit normal vector to the boundary, and T_x and T_y are the specified boundary tractions.

Essential boundary conditions:

$$u = \bar{u}, \quad v = \bar{v}\tag{4.9}$$

where \bar{u} and \bar{v} are the specified boundary displacements.

2. Weak Form

The weighted residual statement is formed by using the variation of the displacements as weights with the governing differential equations as follows:

$$\begin{aligned}\iint \delta u \left(\frac{\partial \sigma_x}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} + f_x \right) dx dy &= 0 \\ \iint \delta v \left(\frac{\partial \sigma_y}{\partial y} + \frac{\partial \sigma_{xy}}{\partial x} + f_y \right) dx dy &= 0\end{aligned}\tag{4.10}$$

The weak form for the two equations after integrating by parts can be written as

$$\begin{aligned}-\iint \left(\frac{\partial \delta u}{\partial x} \sigma_x + \frac{\partial \delta u}{\partial y} \sigma_{xy} \right) dx dy + \int \delta u T_x dS + \int f_x \delta u dx dy &= 0 \\ -\iint \left(\frac{\partial \delta v}{\partial y} \sigma_y + \frac{\partial \delta v}{\partial x} \sigma_{xy} \right) dx dy + \int \delta v T_y dS + \int f_y \delta v dx dy &= 0\end{aligned}\tag{4.11}$$

The two equations can be combined into one, since δu and δv are independent, to obtain

$$-\iint \left(\delta \varepsilon_x \sigma_x + \delta \varepsilon_{xy} \sigma_{xy} + \delta \varepsilon_y \sigma_y \right) dx dy + \int (\delta u T_x + \delta v T_y) dS + \int (f_x \delta u + f_y \delta v) dx dy = 0\tag{4.12}$$

3. Discretized Weak Form

Once the weak form is calculated, the primary variables u and v are approximated using interpolation functions. In conventional finite elements, the interpolation functions

used for approximation of the variables are of the Lagrange type. The variables u and v are approximated using n interpolation functions N_i as:

$$u = \sum_{j=1}^n u_j N_j \quad v = \sum_{j=1}^n v_j N_j \quad 4.13$$

The general displacement vector for each element can be represented as:

$$q = [u_1, v_1, u_2, v_2, \dots, u_n, v_n] \quad 4.14$$

Substituting the approximations for the variations of u and v into the weak form, we obtain the following equation:

$$\sum_{i=1}^{2n} \delta q_i \left[\iint \left\{ \left(f_x \frac{\partial u}{\partial q_i} + f_y \frac{\partial v}{\partial q_i} \right) - \left(\frac{\partial \varepsilon_x}{\partial q_i} \sigma_x + \frac{\partial \varepsilon_{xy}}{\partial q_i} \sigma_{xy} + \frac{\partial \varepsilon_y}{\partial q_i} \sigma_y \right) \right\} dx dy + \int \left(\frac{\partial u}{\partial q_i} T_x + \frac{\partial v}{\partial q_i} T_y \right) dS \right] = 0 \quad 4.15$$

where i ranges from 1 to $2n$.

Since the δq_i are arbitrary and independent, the same equation can be written as:

$$\int f_x \frac{\partial u}{\partial q_i} dx dy + \int \frac{\partial u}{\partial q_i} T_x dS + \int f_y \frac{\partial v}{\partial q_i} dx dy + \int \frac{\partial v}{\partial q_i} T_y dS - \sum_{\alpha=1}^3 \int \sigma_{\alpha} \frac{\partial \varepsilon_{\alpha}}{\partial q_i} dA = 0 \quad 4.16$$

where, $\sigma_{\alpha} = \begin{bmatrix} \sigma_x \\ \sigma_y \\ \sigma_{xy} \end{bmatrix} \quad \varepsilon_{\alpha} = \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_{xy} \end{bmatrix}$

$$\text{Also, } \sigma_{\alpha} = C_{\alpha\beta} \varepsilon_{\beta} \quad 4.17$$

The terms in equation 4.17 can be written in terms of B matrix defined as $\bar{B}_{\beta i} = \frac{\partial \varepsilon_{\beta}}{\partial q_i}$,

$$\text{Hence, } \varepsilon_{\beta} = \bar{B}_{\beta i} q_i$$

$$\text{and } \sigma_{\alpha} = C_{\alpha\beta} \bar{B}_{\beta i} q_i \quad 4.18$$

Substituting equation 4.18 to equation 4.16, we obtain:

$$[K]\{q\} = \{F\} \quad 4.19$$

$$\text{where, } K_{ij} = \int \bar{B}_{\alpha i} C_{\alpha\beta} \bar{B}_{\beta j} q_j dx dy \quad 4.20$$

Using any general interpolation functions N_i , the strain displacement matrix called the \bar{B} matrix for an element with n interpolation functions can be written as:

$$\bar{B}_{\alpha i} = \frac{\partial \varepsilon_{\alpha}}{\partial q_i} = \left[\begin{array}{cc|cc|cccc} \frac{\partial N_1}{\partial x} & 0 & \frac{\partial N_2}{\partial x} & 0 & & \frac{\partial N_n}{\partial x} & 0 \\ 0 & \frac{\partial N_1}{\partial y} & 0 & \frac{\partial N_2}{\partial y} & \dots\dots & 0 & \frac{\partial N_n}{\partial y} \\ \hline \frac{\partial N_1}{\partial y} & \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial y} & \frac{\partial N_2}{\partial x} & & \frac{\partial N_n}{\partial y} & \frac{\partial N_n}{\partial x} \end{array} \right] \quad 4.21$$

Once the \bar{B} matrix is calculated, it can be used to calculate the stiffness matrix and the force vector for every element.

4. Sub-Parametric Formulation

The calculations required for the \bar{B} matrix and the stiffness matrix for elements with distorted shape are very complex in the physical coordinate system and are not practically possible in all cases. To be able to use curved edges for an element, the element properties are calculated using a local coordinate system. The transformation from the global to local coordinate system can be done using isoparametric, subparametric and superparametric formulations.

Isoparametric formulation involves the use of the same interpolation functions for approximating the geometry as well as the field variable. In the case of subparametric formulation, the polynomial function used to define the geometry has a lower order than the field variable. i.e a lesser number of nodes are used to describe the geometry as compared to the displacement. In the case of elements with curved edges used for arbitrary shape, numerical integration is needed since the exact integration is not practical.

The subparametric formulation involves the use of lower order polynomials for geometry approximation. For example, the displacement u and v are approximated using interpolation functions N_i , defined in the local coordinate system (ξ, η) as

$$u = \sum_{i=1}^n u_i N_i(\xi, \eta) \quad v = \sum_{i=1}^n v_i N_i(\xi, \eta) \quad 4.22$$

The physical coordinates (x, y) are approximated using interpolation functions S_i , which have a lower order than N_i as

$$x = \sum_{i=1}^m x_i S_i(\xi, \eta) \quad y = \sum_{i=1}^m y_i S_i(\xi, \eta) \quad m < n \quad 4.23$$

In calculation of the element properties for the global coordinate system, but working in the local coordinate system, there are two main challenges.

- a) Global derivatives
- b) Differential area

The next two sections discuss how these two challenges have been addressed in this work.

5. Global Derivatives

The relationship between the global derivatives of the interpolation functions N_i and the local derivatives can be calculated by using the chain rule of derivatives:

$$\begin{bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} \begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{bmatrix} \quad 4.24$$

The coefficient matrix in equation 4.24 is termed as the Jacobian matrix, J .

$$J \equiv \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} \quad 4.25$$

The terms in the Jacobian matrix can be calculated by approximating the physical coordinates (x, y) with interpolation functions S_i .

$$x = \sum_{i=1}^m x_i S_i(\xi, \eta) \quad y = \sum_{i=1}^m y_i S_i(\xi, \eta) \quad 4.26$$

The partial derivatives of the physical coordinates with respect to the local coordinates, can be calculated directly since the S_i are functions of the local coordinates.

$$\begin{aligned}\frac{\partial x}{\partial \xi} &= \sum_{i=1}^m x_i \frac{\partial S_i(\xi, \eta)}{\partial \xi} & \frac{\partial y}{\partial \xi} &= \sum_{i=1}^m y_i \frac{\partial S_i(\xi, \eta)}{\partial \xi} \\ \frac{\partial x}{\partial \eta} &= \sum_{i=1}^m x_i \frac{\partial S_i(\xi, \eta)}{\partial \eta} & \frac{\partial y}{\partial \eta} &= \sum_{i=1}^m y_i \frac{\partial S_i(\xi, \eta)}{\partial \eta}\end{aligned}\tag{4.27}$$

The global derivatives are calculated by multiplying the local derivatives with the inverse of the Jacobian matrix.i.e.

$$\begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{bmatrix} = J^{-1} \begin{bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \end{bmatrix}\tag{4.28}$$

6. Differential Area

The differential area $dxdy$ needs to be expressed in terms of $d\xi d\eta$ in order to integrate in the local coordinate system. The differential area transformation can be expressed in terms of the Jacobian as follows [34]:

$$dxdy = |J| d\xi d\eta\tag{4.29}$$

This transformation helps in calculating the integrand in the local coordinate system. This integrand can then be solved using the Gauss Legendre's method of numerical integration. Each term is calculated at each quadrature point and then summed up to calculate the value of the integral.

7. B-spline Finite Elements

This implementation of B-spline finite elements involves the approximation of the primary variables using cubic order 2D B-spline interpolation functions. The B-spline functions for a two dimensional region are calculated using the tensor product of the 1D functions as discussed in the earlier section. The functions are defined individually for

separate regions called patches as shown in fig.14. Each patch has 16 interpolation functions associated with it.

The approximation for each patch is defined separately because of the local support for each function. This approximation strategy increases the computational efficiency by allowing the integration of only the non-zero values involved in the integral. In this method, the general formulae for calculating the B matrix, stiffness matrix and the force vector are the same as for conventional interpolation, except for the change in the interpolation functions and the regions of integration. The displacement components u and v are approximated by B-spline interpolation of cubic order for each patch. The geometry is approximated using the quadratic Lagrange interpolation functions. The approximation of u and v for each patch are written as:

$$u = \sum_{i=0}^{15} u_i B_i(t_1, t_2) \quad v = \sum_{i=0}^{15} v_i B_i(t_1, t_2) \quad 4.30$$

The interpolation functions B_i are defined in the knot coordinate system. The u_i and v_i in conventional finite elements represent the nodal degrees of freedom. In the case of B-spline interpolation these represent the degrees of freedom at the control points. At the boundaries of a cluster, the control points are same as the nodes on an element boundary. For each patch, there are 16 interpolation functions, and hence 16 control points, each of which has two degrees of freedom. Hence for each patch, the number of degrees of freedom is 32.

This implementation involves three coordinate systems. The physical coordinate system (x, y) which define the element geometry, the knot coordinate system in which the interpolation functions are defined, and the local coordinate system required to calculate the properties using numerical integration.

The calculation of the element properties in this case requires the calculation of two Jacobians, for transformation from global to knot and knot to local coordinate system. Figure 22 shows the three coordinate systems involved in the calculation of the integrals for a B-spline element.

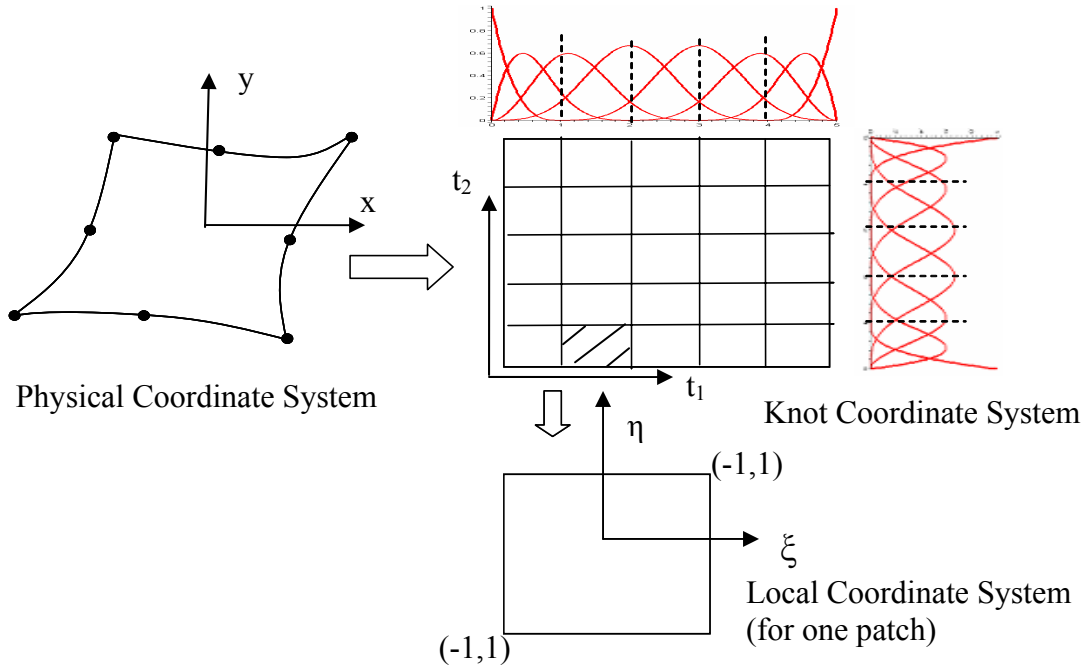


Fig. 22. Three coordinate system for a 2D B-spline cluster.

The element properties can be calculated by substituting the B-spline interpolation functions in the standard formulae for the \bar{B} matrix, stiffness matrix used in the conventional finite element. These standard formulae require the calculation of the global derivatives of interpolation functions and the differential area.

Global Derivatives

The \bar{B} matrix is calculated from the formula given in equation 4.21. Since the number of interpolation functions for each patch is 16, n is equal to 16 for each patch. The \bar{B} matrix calculation requires the global derivatives of the interpolation functions B_i . The interpolation functions are defined in the knot coordinate system (t_1, t_2) . The calculation of the global derivatives requires the transformation from the physical coordinate system to the knot coordinate system. The global derivatives are calculated using the chain rule shown in equation 4.24. The only difference is in the coordinate systems, in this case the physical coordinate system is transformed into the knot coordinate system to calculate the Jacobian given in equation 4.25.

The terms in the Jacobian matrix are calculated by approximating the physical coordinates using Lagrange interpolation. The physical coordinates are approximated using Lagrange interpolation defined in the knot coordinate system (t_1, t_2) as shown in equation 4.31:

$$x = \sum_{i=0}^7 x_i N_i(t_1, t_2) \quad y = \sum_{i=0}^7 y_i N_i(t_1, t_2) \quad 4.31$$

Hence, $\frac{\partial x}{\partial t_1} = \sum_{i=0}^7 x_i \frac{\partial N_i(t_1, t_2)}{\partial t_1}$. Similarly other derivatives can be calculated,

which are further required to calculate the Jacobian matrix from the first transformation.

The N_i are Lagrange interpolation functions defined in the knot coordinate system (t_1, t_2) calculated by the transformation shown in fig. 23. In this transformation, the physical coordinates of the entire cluster are represented by 8 nodes. These nodes are mapped to the nodes in the knot coordinate system depending on the connectivity specified for the element mesh as in the case of the conventional finite elements. For example, the a^{th} node in the physical coordinate system would correspond to the a^{th} node in the knot coordinate system. These nodes do not have any degrees of freedom associated with them, and are just used to represent the geometry of the element. In the connectivity matrix, there is one set of nodes to which the degrees of freedom are assigned, and one set to which the physical coordinates are assigned. The nodes to which the degrees of freedom are associated are just used to calculate the degree of freedom map for each cluster, and do not contribute in the geometry transformation of the cluster. Hence in the connectivity matrix for each element, the first eight nodes are just the geometric nodes with no degrees of freedom associated with them, and the subsequent nodes have 2 degrees of freedom per node, but do not have any physical coordinates associated with them.

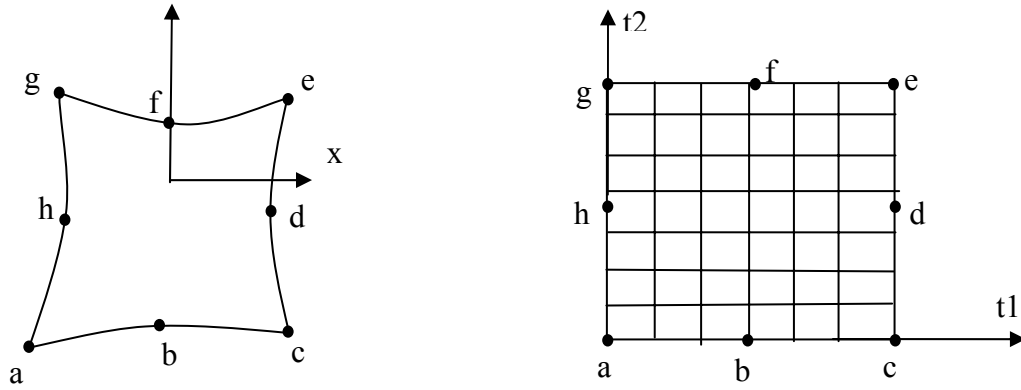


Fig. 23. Coordinate mapping from physical coordinates to knot coordinates.

Differential Area

Once the Jacobian from the transformation from physical coordinate system to knot coordinate system is calculated, the differential area is required to calculate the element properties. The integrand for each patch is now calculated in the local coordinate system for numerical integration. The coordinates of each patch are mapped into the local coordinate system as shown in fig. 24 using the Lagrange interpolation used in conventional finite elements. The order or local numbering in this transformation always remains the same. The sequence is the same as shown in the figure.

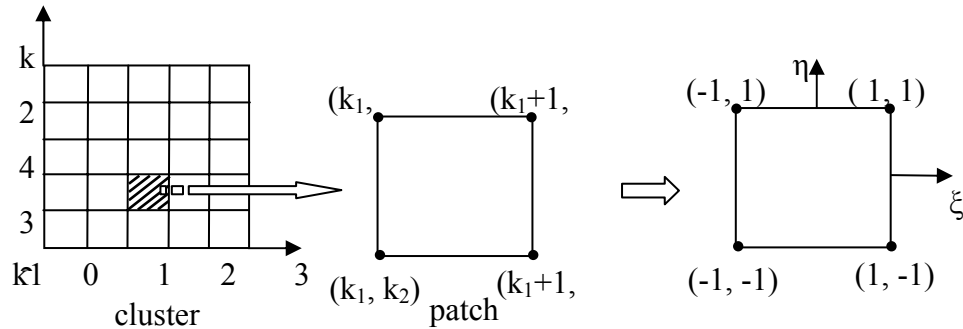


Fig. 24. Transformation from knot coordinates to local coordinates for each patch in a cluster.

To calculate the differential area, $dx dy$ is transformed to be in terms of $d\xi d\eta$. This calculation is a two step procedure involving two transformations. First, the physical coordinates (x,y) are transformed to the knot coordinate system (t_1, t_2) , and then the knot coordinate system is transformed to the local coordinate system (ξ, η) . The jacobian obtained from the first transformation i.e for coordinate mapping from physical coordinate system to knot coordinate system is termed as J_1 , and is calculated using equation 4.31.

$$\int_{x_1}^{x_2} \int_{y_1}^{y_2} F(x, y) dx dy = \iint_{t_1 t_2} F(x(t_1, t_2), y(t_1, t_2)) J_1 dt_1 dt_2 \quad 4.32$$

For the second transformation t_1 and t_2 are approximated using the standard Lagrange interpolation functions used in conventional finite elements.

$$t_1 = \sum_{i=0}^7 t_1^i S_i(\xi, \eta) \quad t_2 = \sum_{i=0}^7 t_2^i S_i(\xi, \eta) \quad 4.33$$

The values of the t_1 and t_2 are calculated at each quadrature point, and then substituted in the \bar{B} matrix to get the stiffness matrix. The transformation of the differential area to the local coordinate system is done by calculating the second Jacobian as:

$$J_2 = \begin{bmatrix} \frac{\partial t_1}{\partial \xi} & \frac{\partial t_2}{\partial \xi} \\ \frac{\partial t_1}{\partial \eta} & \frac{\partial t_2}{\partial \eta} \end{bmatrix} \quad 4.34$$

The terms in this Jacobian are calculated from the approximation of the knot coordinates by using Lagrange interpolation functions (equation 4.34). The derivatives can be calculated using the same approximation as follows:

$$\begin{aligned} \frac{\partial t_1}{\partial \xi} &= \sum_{i=1}^m t_1^i \frac{\partial S_i(\xi, \eta)}{\partial \xi} & \frac{\partial t_2}{\partial \xi} &= \sum_{i=1}^m t_2^i \frac{\partial S_i(\xi, \eta)}{\partial \xi} \\ \frac{\partial t_1}{\partial \eta} &= \sum_{i=1}^m t_1^i \frac{\partial S_i(\xi, \eta)}{\partial \eta} & \frac{\partial t_2}{\partial \eta} &= \sum_{i=1}^m t_2^i \frac{\partial S_i(\xi, \eta)}{\partial \eta} \end{aligned} \quad 4.35$$

Hence equation 4.33 can now be written as a function of the local coordinates (ξ, η)

$$\int_{x_1}^{x_2} \int_{y_1}^{y_2} F(x, y) dx dy = \int_{-1}^1 \int_{-1}^1 F(x(t_1(\xi, \eta), t_2(\xi, \eta)), y(t_1(\xi, \eta), t_2(\xi, \eta))) J_1 J_2 d\xi d\eta \quad 4.36$$

The integrand is calculated in terms of the local coordinate system (ξ, η) and then the properties are calculated at each quadrature point for each patch. The patch properties are then calculated using numerical integration. These properties are then used to calculate the properties for a cluster, and subsequently for the entire model.

8. Assembly of Stiffness Matrix

The element properties including the stiffness matrix and the force vector for each patch are calculated as described above. To solve for the equations for the entire structure, the properties for each patch are assembled using the compatibility and equilibrium conditions as in the case of conventional finite element. To be able to assemble the equations, the degree of freedom map for each element is required. The only difference in this case is that the assembly is a two step procedure. First, the degree of freedom map for each patch is calculated and then the degree of freedom map for the entire structure is calculated to get the overall equations.

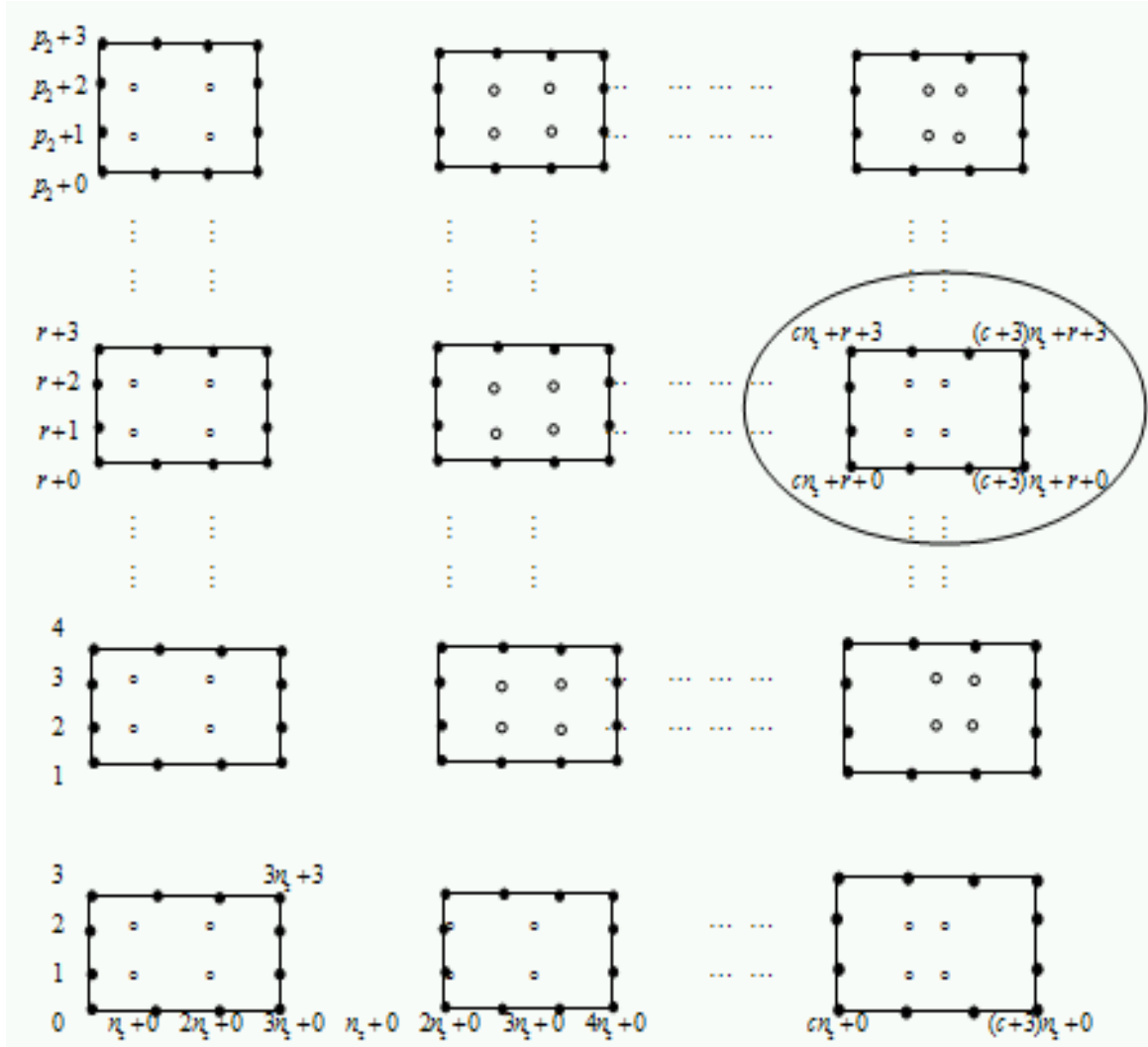


Fig. 25 Renumbering of the control points for a cluster. $n_2 = p_2 + 4$
 marked patch shows numbering for patch in the c^{th} column and r^{th} row.

Once the properties for each patch are calculated, the stiffness matrix for the entire cluster is assembled using the degree of freedom map for each patch. The degree of freedom map for each cluster is calculated automatically in a subroutine, based on the number of interpolation functions that are shared in between different patches. In this subroutine, the control points are renumbered for a cluster using the overlapping functions. The renumbering of cluster depends on the location of the patches in the cluster and the total number of patches in a cluster as can be seen from fig. 25. The

location of a particular patch in the cluster can be designated by r and c , where r is the patch number in the t_2 direction and c is the patch number in the t_1 direction. The total number of patches in the t_2 direction is specified as p_2 . Figure 26 shows a cluster with $p_2 = 4$, and shows different patches with the r and c labeled. Figure 25 shows the numbering of different control points associated with different patches in a cluster, according to the location of the patch in the cluster, and the total number of patches in a cluster. The patch with $c = 0, r = 0$, shares 12 control points with the patch with $c = 0, r = 1$ as can be seen in fig. 25. This is seen due to the overlap of the interpolation functions in between two patches. The figure also shows the formula for a general patch with the number of patches in t_2 as r and the number of patches in t_1 direction as c . Figure 27 shows the control point numbering for the cluster shown in fig.26 after using the control point map illustrated in fig.25. The degree of freedom associated with each control point can be calculated accordingly as there are two degrees of freedom per control point.

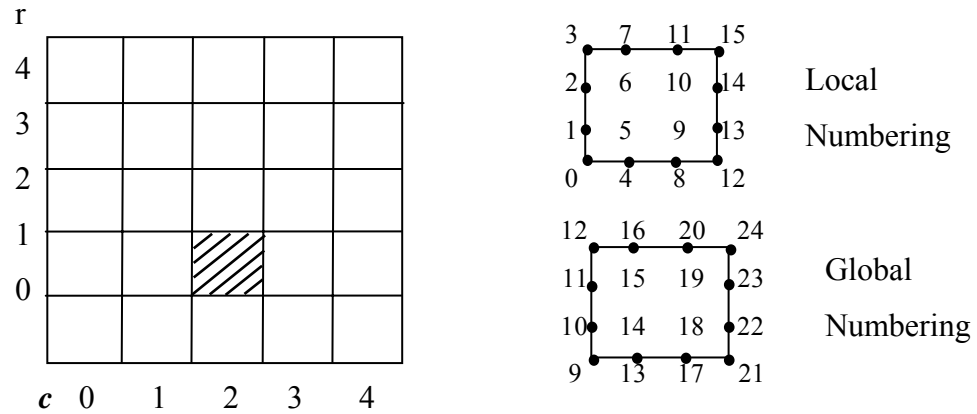


Fig.26 Control point map for a patch.

With the help of the control points for a cluster, the matrices for each cluster are calculated. These matrices are then assembled into global matrices using the shared boundaries between clusters, as in the case of conventional finite elements. In this case

there is no overlap of control points between clusters, and only the boundary degree of freedom are assembled together. Figure 26 shows a cluster with the internal and the boundary control points. The boundary control points are the only ones shared between two clusters as shown in the figure. The internal degrees of freedom do not have to be necessarily related to any physical coordinates as they are not used in the geometry modeling or assembly.

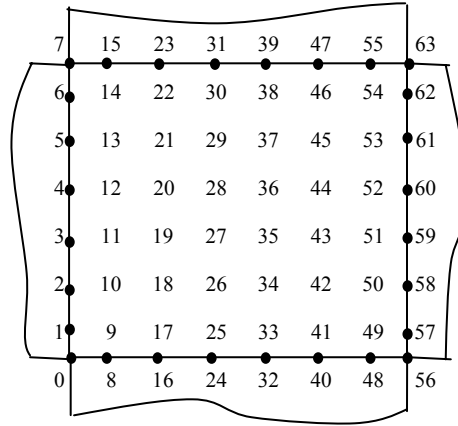


Fig.27 Control points numbering for a cluster with 8 control points on each side t_1 and t_2 .

Once the global properties are calculated, the boundary conditions are applied similar to conventional finite element method. The equations are solved for the unknown displacements at the control points. These displacements are used to calculate the stress at the quadrature points for each patch.

CHAPTER V

COMPARISON OF THE B-SPLINE ELEMENTS WITH SERENDIPITY ELEMENTS

This section describes the two dimensional elasticity problems considered to validate the B-spline finite element method. To assess the validity of the method, the test cases for some particular 2d plane elasticity problems have been considered. The solutions were compared with the conventional finite element solutions using 8-noded serendipity elements [35]. The experiments were carried out for two problems: a single lap shear specimen and a plate with hole subject to in-plane tension loading. The objective of these experiments is to validate the method by comparing the results with a reference solution obtained using conventional finite elements and a very refined mesh. The solutions are further compared to study the convergence behavior and the efficiency for both the methods.

1. Single Lap Shear Joint

Adhesive bonded joints are extensively used in joining parts made of composite materials. There are various methods that are used to measure the strength of the lap joint. The single lap shear test (ASTM D5868) [36], as shown in fig. 28, is widely used to evaluate the strength of the lap joint. These tests give an estimate of the average strength, but cannot provide any idea about the stress concentrations that develop due to the geometry and the difference in the material properties of the adherend and the adhesives. The finite element method can provide a better idea about the stress distribution in the specimen.

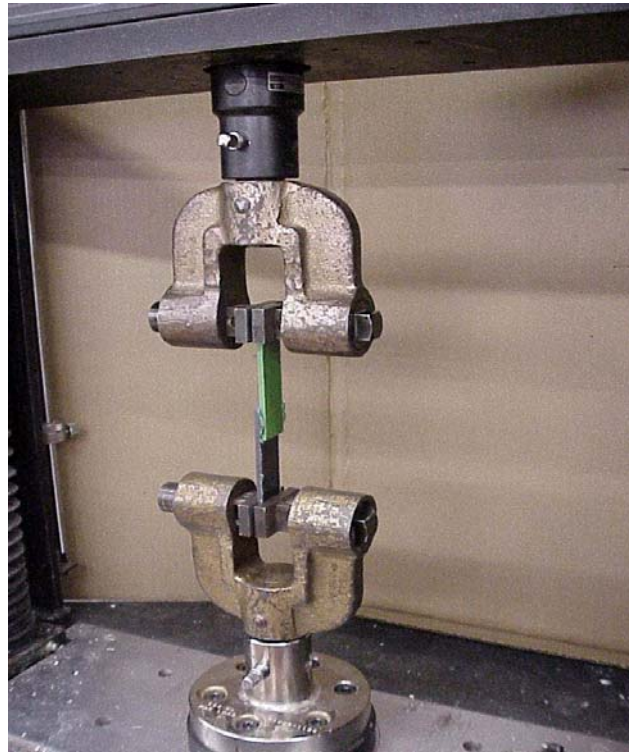


Fig. 28 Lap shear test configuration.

This practical problem has been chosen to compare the two different interpolations used for the finite element method. The stress contours for the two different models will be studied in this section. The variation of the peel stress and the shear stress at the interface between the adhesive and adherend is further studied. The stress values obtained from the two different methods are compared by calculating the error in the stress along the centerline of the adhesive for different number of degrees of freedom. A comparison of the convergence behavior for both the methods with respect to the number of degrees of freedom and the time required to obtain a solution is done.

A two dimensional finite element analysis of an adhesively bonded single lap joint in tension was conducted using both the conventional 8 node Lagrange elements and cubic B-spline elements. Parametric studies to see the effect in the stress variation with the increase in the number of degrees of freedom were performed. The elements are assumed to be plane strain elements, assuming no variation in the width direction.

The bond between adhesive and the adherend is assumed to be perfect and free of defects. The configuration for the model is obtained from the ASTM D5868 standard. The geometry and the boundary conditions are as shown in fig. 29. The boundary conditions are applied such that there is a tensile load acting on the specimen similar to the ASTM standard specifications. A uniformly distributed load is applied at the right end.

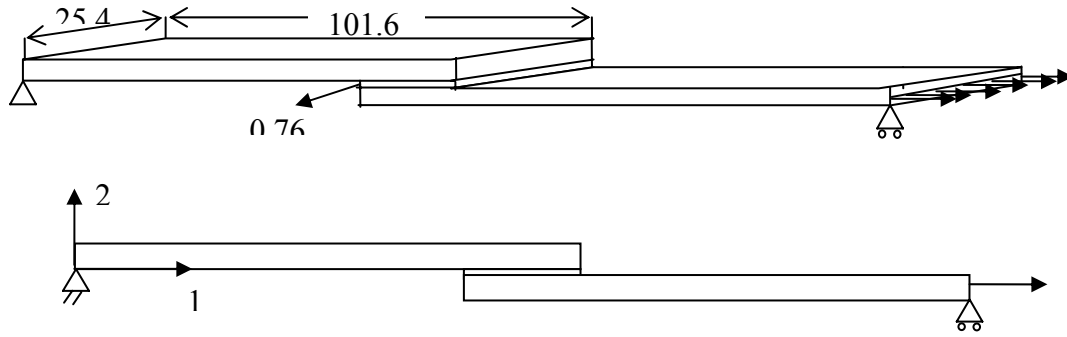


Fig. 29 Single lap joint configuration.

Material Properties

The adherends for the lap joint are made of 16 unidirectional glass epoxy laminas. The properties of the glass fiber are assumed to be $E = 70\text{GPa}$, $\nu = 0.2$ and the epoxy matrix properties are assumed to be $E = 3.5\text{ GPa}$ and $\nu = 0.35$. The fiber volume fraction is assumed to be 0.6. The thickness of each lamina is 0.17272 mm and the stacking sequence is $[0]_{16}$. The lamina properties obtained using the rule of mixtures is calculated as:

$$E_{11} = 43.4\text{ GPa} \quad E_{22} = E_{33} = 14.79\text{ GPa}$$

$$G_{12} = G_{13} = 4.45\text{ GPa} \quad \nu_{12} = \nu_{13} = 0.26$$

The adhesive considered is a structural adhesive, FM-73. It is considered to be a linear elastic isotropic material with the properties given as

$$E = 1.8\text{GPa} \quad \nu = 0.4.$$

A series of models were analyzed to study the convergence behavior in both the conventional finite element and B-spline finite element. The cluster boundaries for the

B-spline mesh are as shown in fig. 30. The geometry for each of these clusters is represented using 8 nodes, and for a graded mesh the mid-side nodes are biased towards the area of interest. There are no degrees of freedom associated with these 8 nodes. They are only used to represent the geometry. The degrees of freedom associated with each cluster can be specified separately during run time. Several experiments with a varying number of degrees of freedom for each cluster have been conducted.

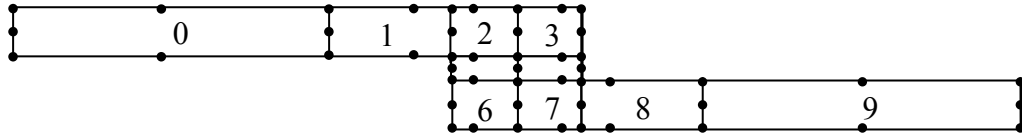


Fig. 30 Cluster mesh for B-spline finite elements.

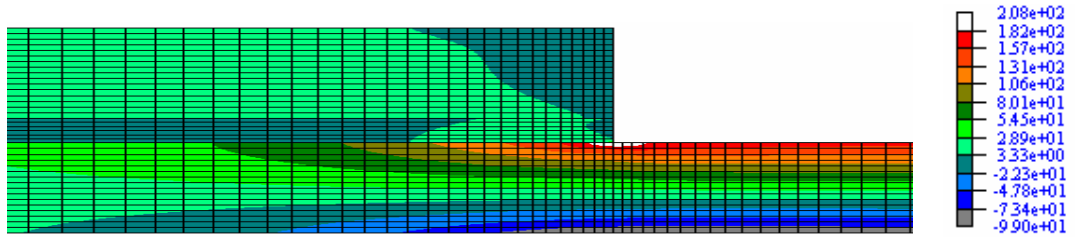


Fig. 31 Deformed shape for the lap shear specimen.

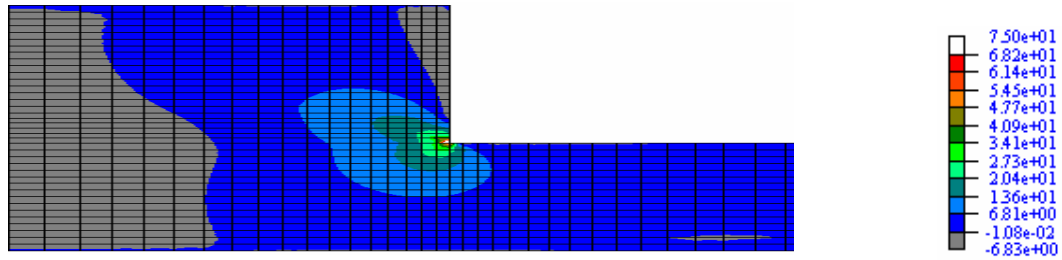
The typical deformed shape for the specimen as obtained for a very refined mesh using conventional finite element solution is as shown in fig. 31. The displacement magnification scale factor for the deformed shape as shown in fig. 31 is 5. This solution obtained by a very refined mesh of 34714 degrees of freedom is considered as the reference solution to evaluate the B-spline finite element solutions and the less refined conventional finite element solutions.

The stress contours for the three components σ_{xx} , σ_{yy} , σ_{xy} as obtained from a very refined conventional finite element mesh are as shown in fig. 32. The shear stress at the interface is zero in the middle, but peak up at the edges. At the corner of the interface, a very high peel stress and shear stress are observed as can be seen from the figure. For the converged solution, σ_{22} , σ_{12} are continuous in adherend and the adhesive. But in the case

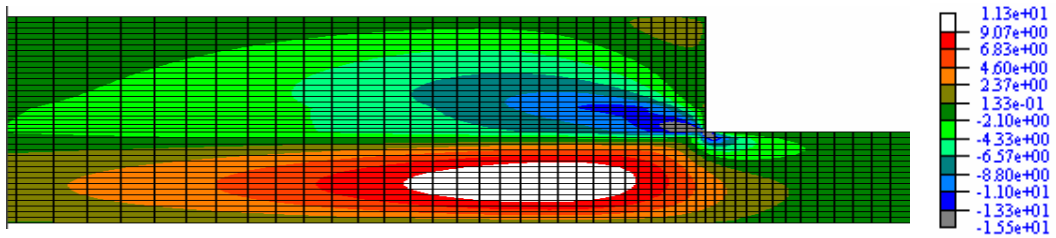
of the less refined mesh, the values at the same point for the two materials differ, and is larger in the adhesive.



a) σ_{11}



b) σ_{22}



c) σ_{12}

Fig. 32 Contour plots of σ_{11} , σ_{22} , σ_{12} obtained by 8 node serendipity elements.

Figure 33 shows a plot of the peel stress at the two interfaces between the adhesive and the adherend layer. The stresses for the lower and upper interfaces are plotted separately for the adhesive and the adherend. As can be seen from the plot the stresses at the middle are approximately zero, with very high peaks at the corners. A

similar plot obtained for the shear stress is shown in fig. 34. The shear stress also attains very high value near the edges, with the maximum shear stress seen in the adhesive.

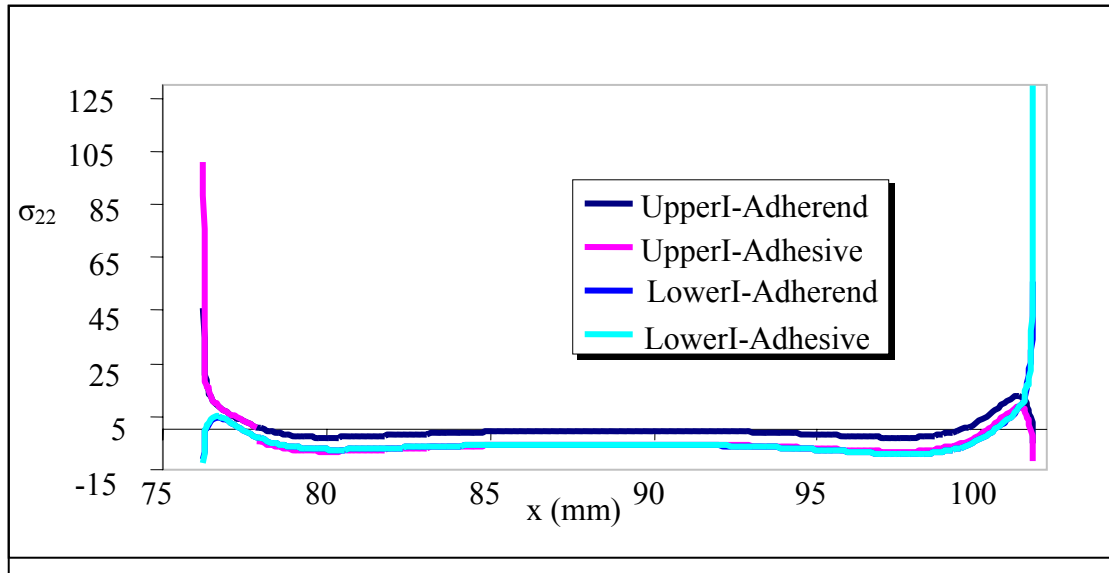


Fig. 33 Variation of peel stress at the interface between the adhesive layer and the adherends.

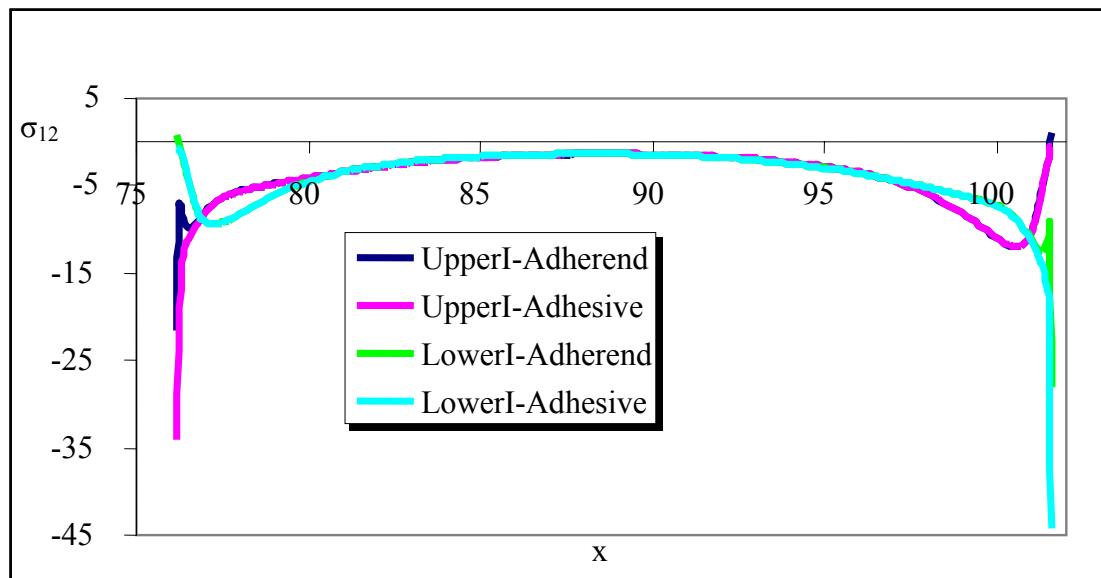


Fig. 34 Variation of shear stress at the interface between the adhesive layer and the adherends.

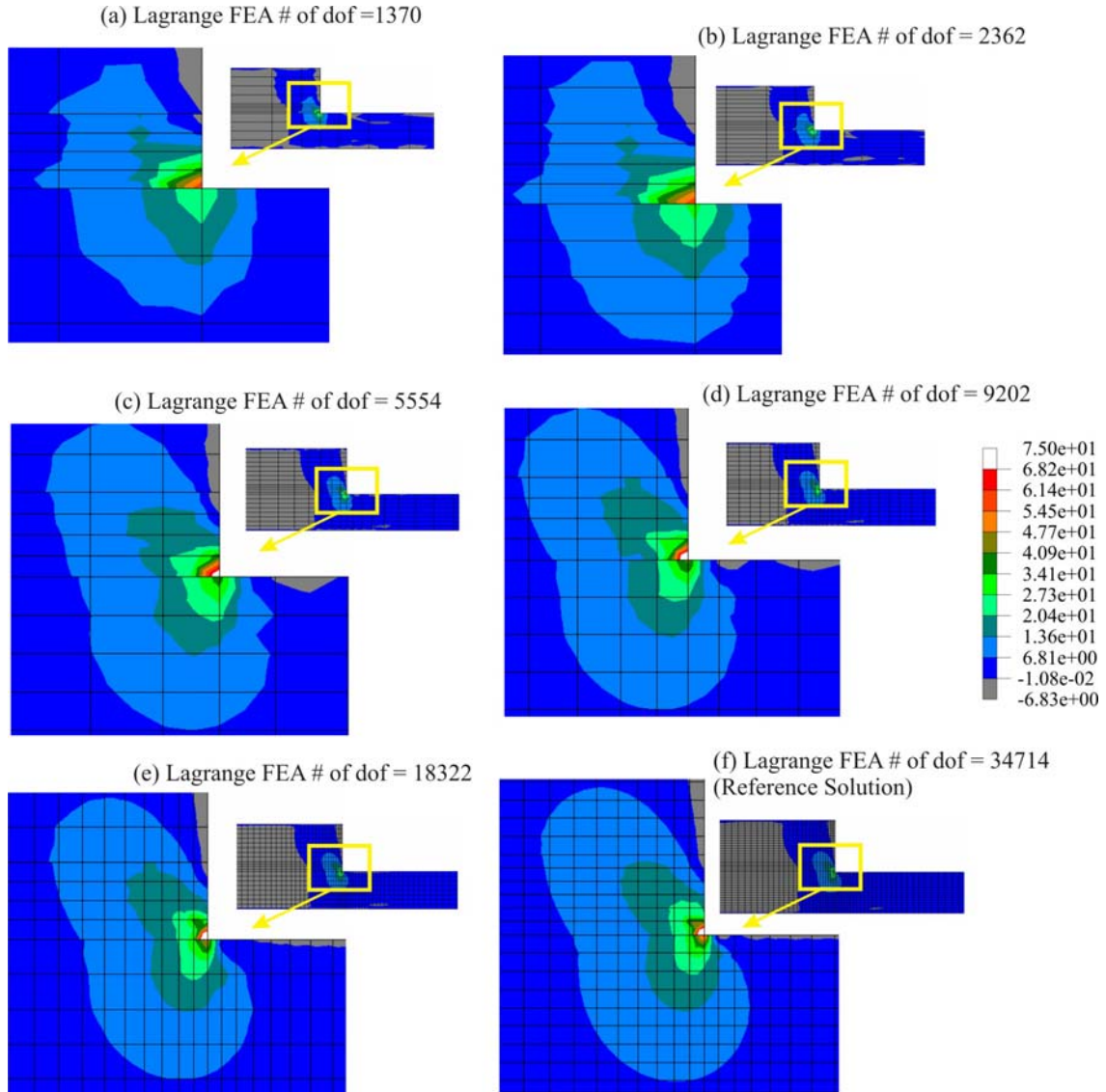


Fig. 35 Convergence behavior for peel stress near the interface for conventional finite element solution.

The results obtained from the finite element solution are taken as a benchmark to compare the accuracy and study the convergence behavior of the B-spline finite element solution. Several cases for conventional finite elements using Lagrange interpolation for different number of degrees of freedom were run. Figure 35 shows the contour plots of the peel stresses obtained from the conventional solution for different numbers of degrees of freedom. The results are plotted starting from 1370 degrees of freedom till the

reference solution which has 34714 degrees of freedom. As can be seen from the figure, for the case of 1370 and 2362 degrees of freedom the stress plots are not very continuous and do not agree well with the reference solution. Even with an increase to 5554 degrees of freedom, there is an improvement in the results, but they are still not very accurate. The results seem to match the reference solution well in the case of 18322 degrees of freedom.

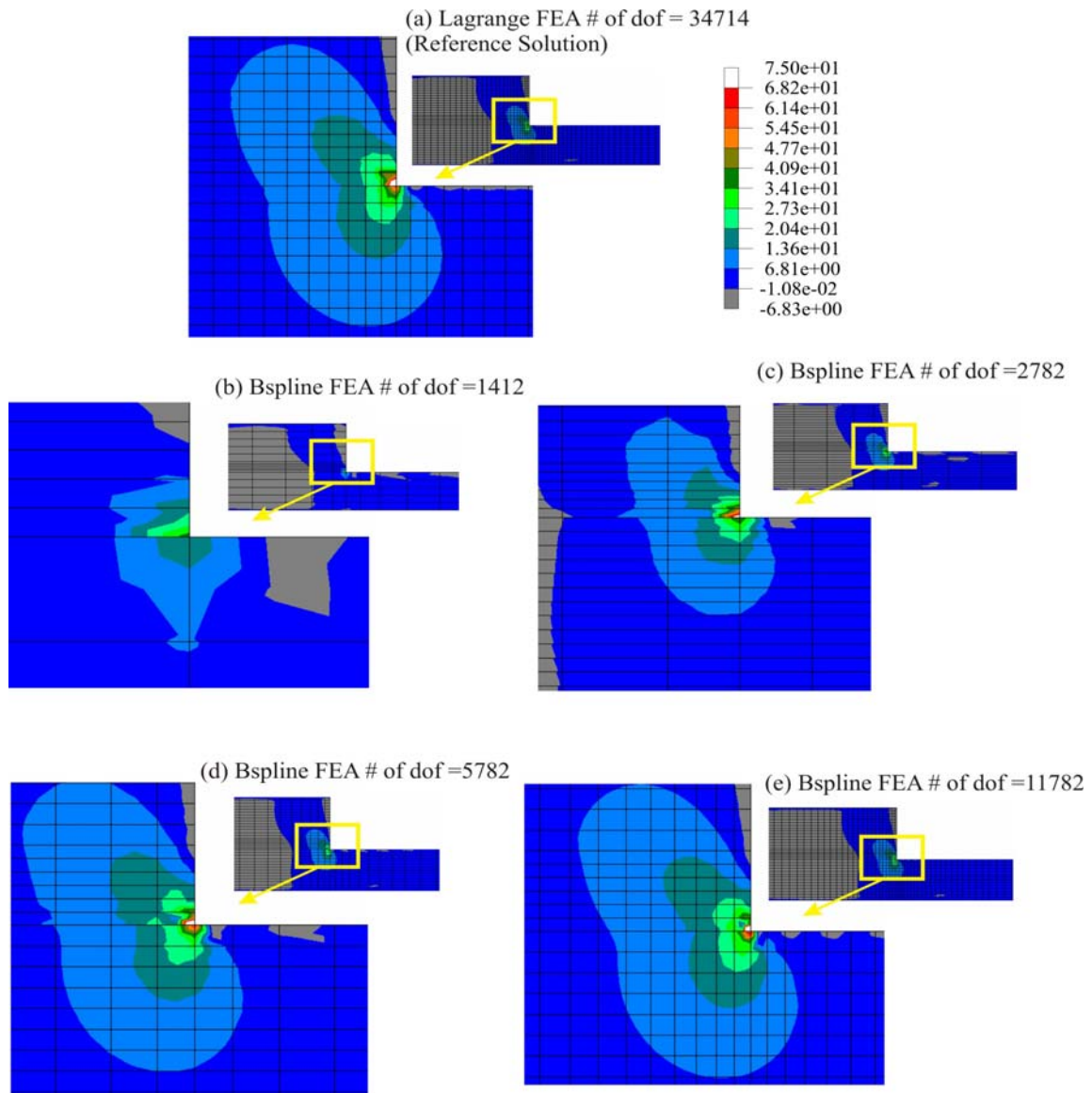


Fig. 36 Convergence behavior for peel stress near the interface using B-spline elements.

A similar study was conducted using B-spline finite elements. The results are as shown in fig. 36. The initial plot shows the case of 1412 degrees of freedom, which does not seem to match well with the reference solution. As the number of degrees of freedom is increased, there is an improvement in the stress contours seen for the case of 2782 degrees of freedom. The results matched much better to the reference solution for 5782 degrees of freedom in the case of B-spline finite elements, as compared to the a case with similar number of degrees of freedom in the case of conventional finite elements. It can be observed from these plots that the stress contours for B-splines match with the reference solution at much lower degrees of freedom than the conventional method. The convergence rate of the solution for the B-spline finite elements is much faster than the conventional case.

The contour plots only provide a qualitative estimate of the convergence behavior of the two different methods for finite elements. To get a better idea of the convergence behavior, the error in the stress values at the centerline in the adhesive is compared for different cases. A curve fit for the stress values on the centerline is found using spline interpolation for different cases of conventional and B-spline elements. The difference between the fitted curves for different degrees of freedom and the fitted curve for the reference stress values is then compared. Fig. 37 shows the error in σ_{22} with respect to the reference solution for two cases using quadratic Lagrange elements, and two cases with a similar number of degrees of freedom using B-Spline interpolation. As can be seen from the figure, the stress values in the middle are almost converging except for the slight oscillations. Near the two ends there are oscillations seen in both B-Spline and Lagrange interpolation cases.

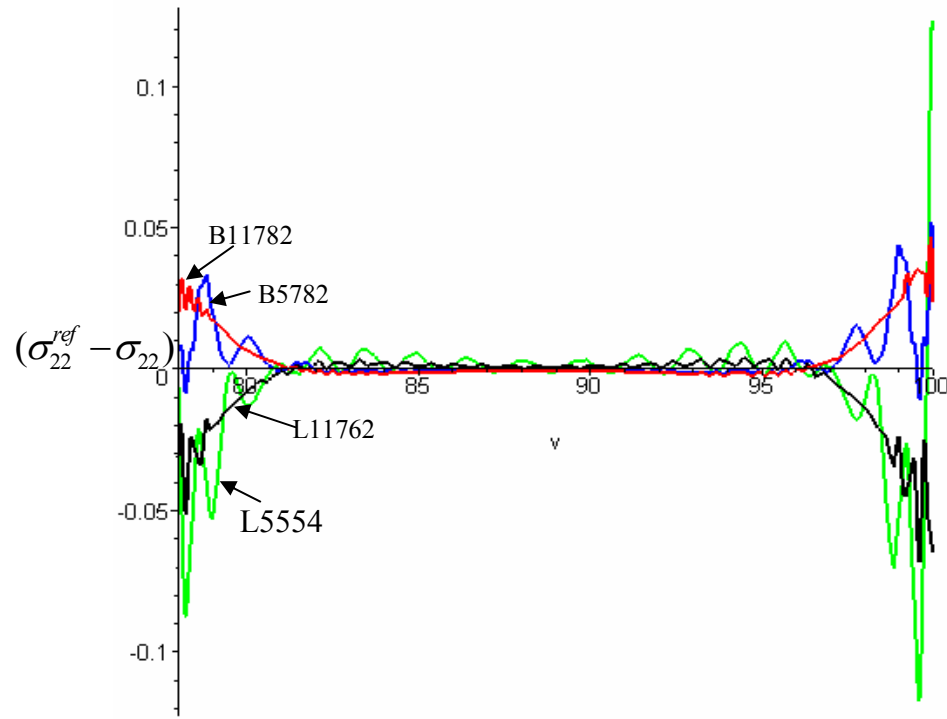


Fig. 37 Difference in σ_{22} value with respect to the reference value for conventional and B-Spline finite elements.

The figure shows higher amplitude of oscillations in the case of Lagrange elements than the B-Spline elements for approximately the same number of degrees of freedom. The figure shows a higher accuracy in the stress values along the centerline in the case of B-Spline finite elements than the conventional finite elements.

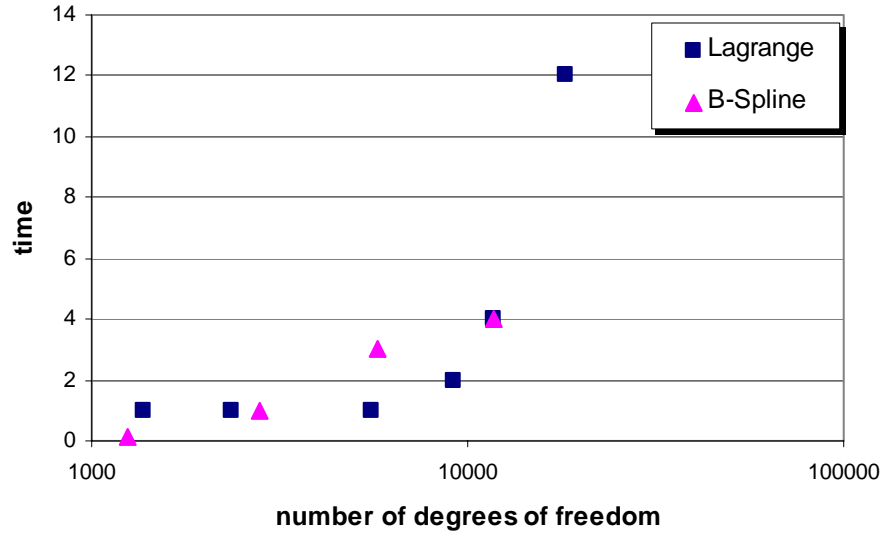


Fig. 38 Solution time vs number of degrees of freedom for conventional and B-Spline finite elements.

A true comparison of the efficiency can be done, if for the same level of accuracy the time required to solve a problem is less in the case of B-spline elements. To compare the time required a plot of time required to solve the equations vs the number of degrees of freedom is shown in fig. 38. As can be seen from the figure, for the same number of degrees of freedom, the solution time is only slightly higher in the case of B-Spline elements. This was expected due to the higher coupling of the stiffness terms of the patches than in the case of conventional finite elements.

To further analyze the accuracy of the two methods with respect to the solution time, an error measure is introduced. The error in the stress values along the centerline obtained from different cases is calculated using the formula:

$$\text{Error} = \sqrt{\int_{x_1}^{x_2} (\sigma_{ref} - \sigma)^2 dx} \quad 5.1$$

This error is calculated by calculating the difference between the curves fitted by spline interpolation. The plot of error in σ_{22} along the adhesive centerline against time is shown in fig. 39. It can be clearly seen from the plot that for the same time to run a

particular case, the error in B-spline finite element is less than the conventional finite elements using quadratic Lagrange interpolation.

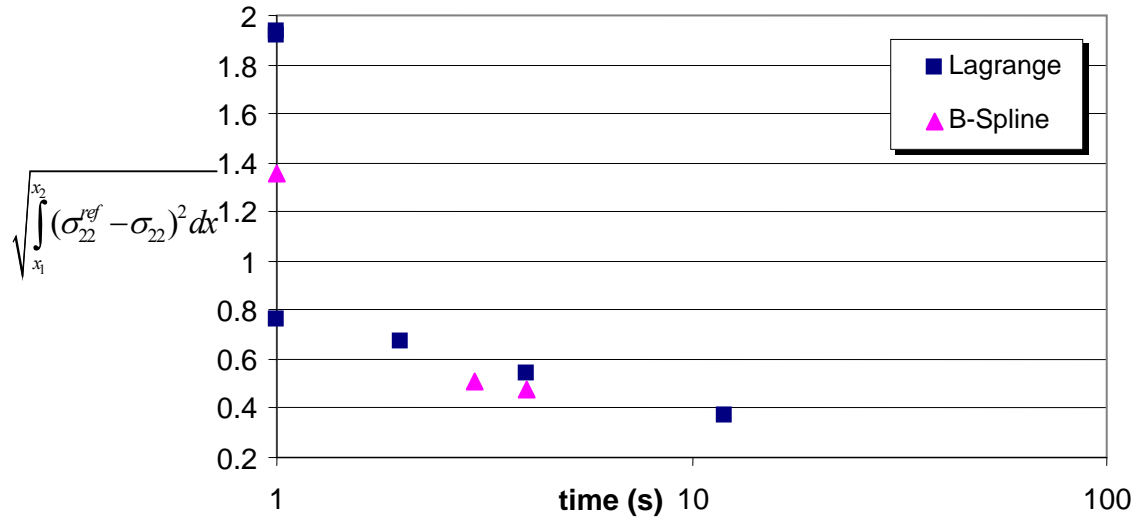


Fig. 39 Error in σ_{22} vs time for both conventional and B-spline finite elements.

A similar study was done for shear stress distribution along the centerline in the adhesive. The shear stress contour plots for conventional finite elements for different number of degrees of freedom are as shown in figure 40. The shear stress distribution near the adhesive adherend interface is shown in these plots, to study the convergence behavior of the stress distribution. A similar study was done for the case of B-spline elements as shown in figure 41. Again in this case, the solution for lower number of degrees of freedom (1412) does not match very well with the very refined mesh reference solution. But as the number of degrees of freedom increase to 2782, the solution matches pretty well with the reference solution.

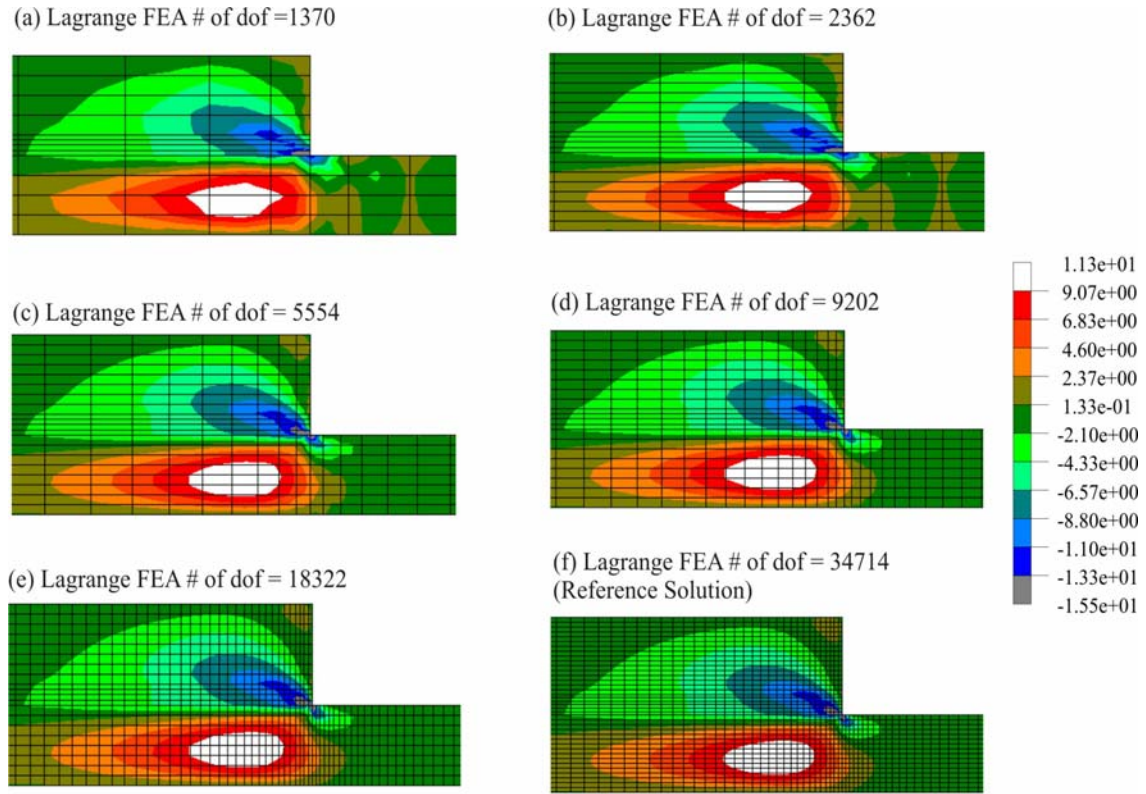


Fig. 40 Convergence behavior for shear stress near the interface using conventional finite element solution.

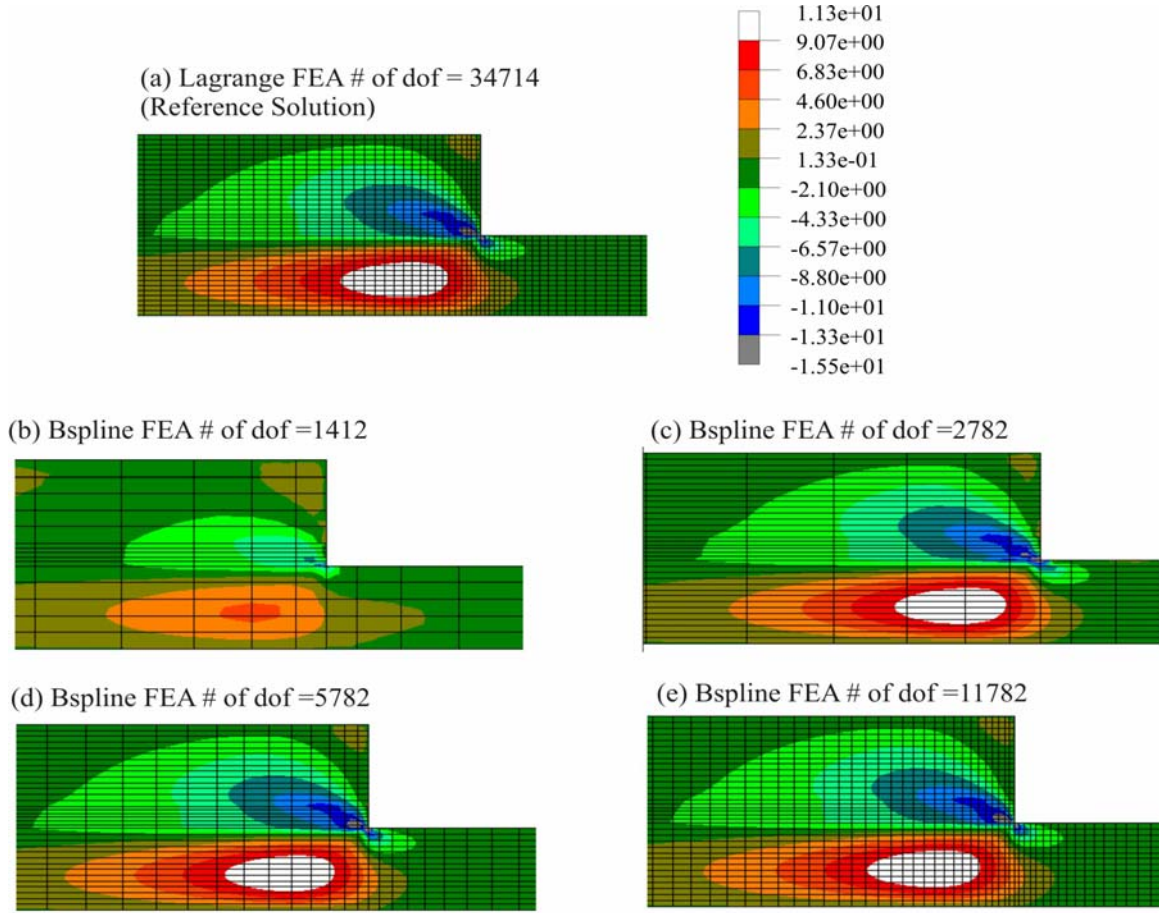


Fig. 41 Convergence behavior for shear stress near the interface using B-spline finite element solution.

For a more quantitative analysis of the convergence behavior in shear stress, the difference in the shear stress with respect to the reference solution is plotted. For all the different cases of degrees of freedom, the data points for stress along the centerline are fitted using spline interpolation. Figure 42 shows the difference between the fitted curve and the reference fitted curve along the centerline for two cases of conventional finite elements and two for the B-spline finite elements. As can be seen from the figure, for almost the same number of degrees of freedom for the two different cases, the convergence behavior is different. The two curves for the Lagrange interpolation case never approach zero, whereas for the case of B-spline interpolation for the middle part,

the difference is almost zero. This plot clearly shows a better convergence rate for the same number of degrees of freedom in the case of B-spline finite elements.

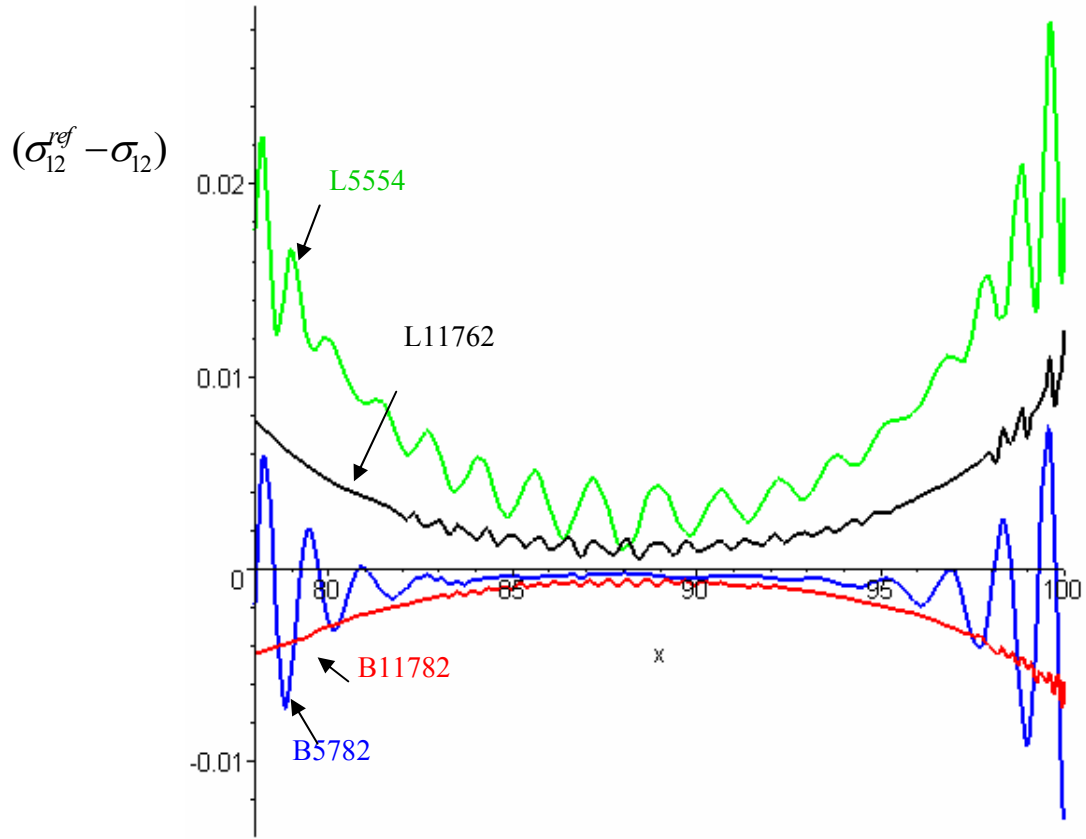


Fig.42 Difference in σ_{12} value with respect to the reference value for conventional and B-Spline finite elements.

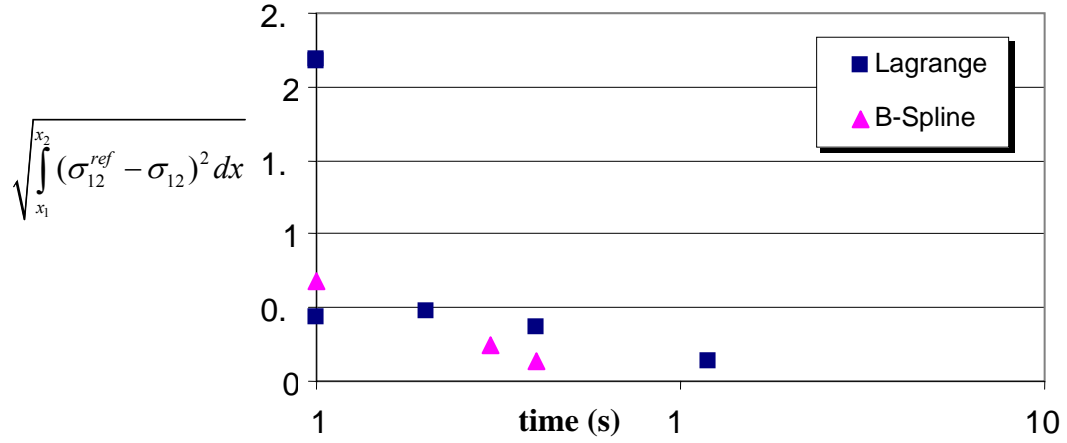


Fig. 43 Error in σ_{12} vs time for both conventional and B-spline finite elements.

The error in the shear component of stress against run time is shown in figure 43. The B-spline finite elements show a much lower error than the conventional elements. For both, the stress components σ_{11} and σ_{12} for the case of lap shear joint, B-spline shows a higher accuracy at lower number of degrees of freedom. Even though the solution time required for the same number of degrees of freedom in B-spline elements is more, the overall time required to attain the same accuracy level is less in the case of B-spline elements.

2. Plate with Elliptical Hole

The second problem considered is a square plate with an elliptical hole, and an applied tensile load in the y direction. The dimensions and the boundary conditions for the plate are as shown in figure 44. Exploiting the symmetry of material, geometry and boundary conditions, only onequarter of the plate is analyzed. The plate is considered to be an aluminum plate, with material properties assumed as:

$$E = 70 \text{ GPa} \quad \nu = 0.33 \quad G = \frac{E}{2(1+\nu)} = 26.3 \text{ GPa}$$

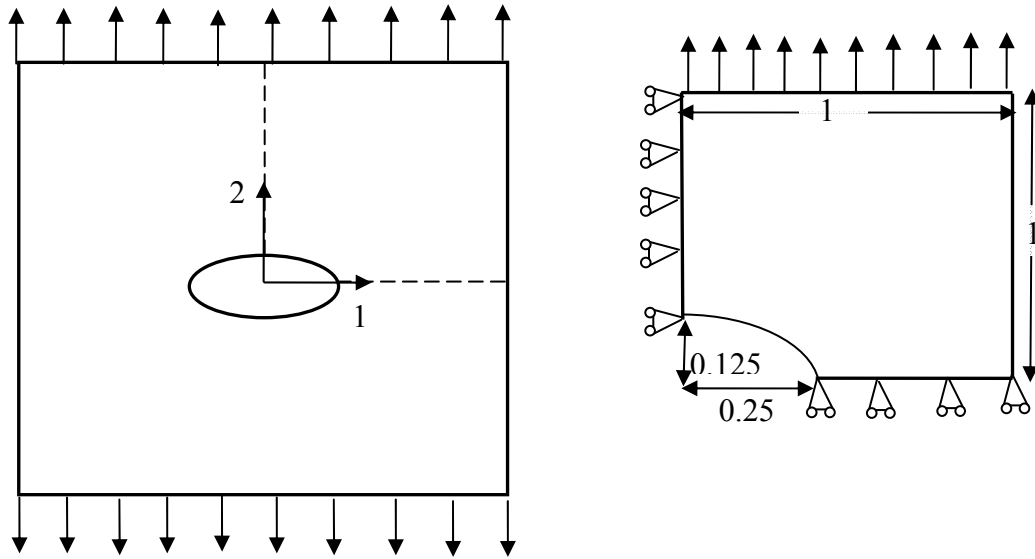


Fig. 44 Plate with elliptical hole.

A two dimensional plane stress analysis was conducted using the two different methods for different number of degrees of freedom. The results from the quadratic Lagrange elements are compared with the cubic B-spline elements by running several experiments with different number of degrees of freedom. The reference solution for comparing the two methods is taken to be a very refined mesh of 150,000 degrees of freedom run in an in-house code using 8 noded Lagrange elements.

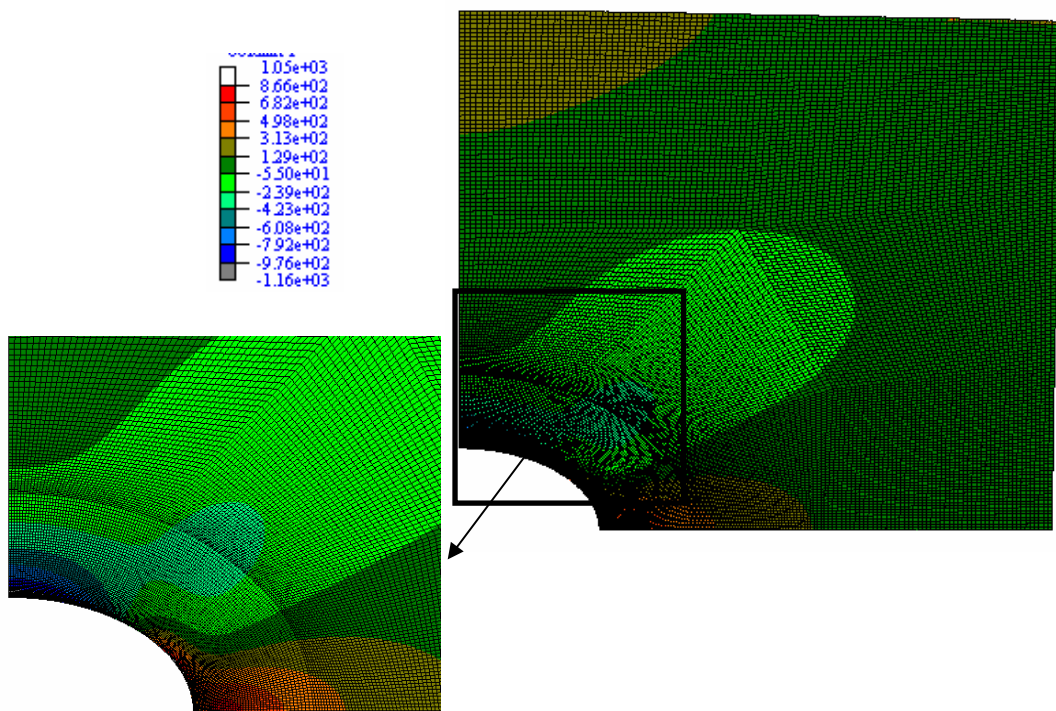


Fig. 45 Deformed quarter plate with contours showing axial stress.

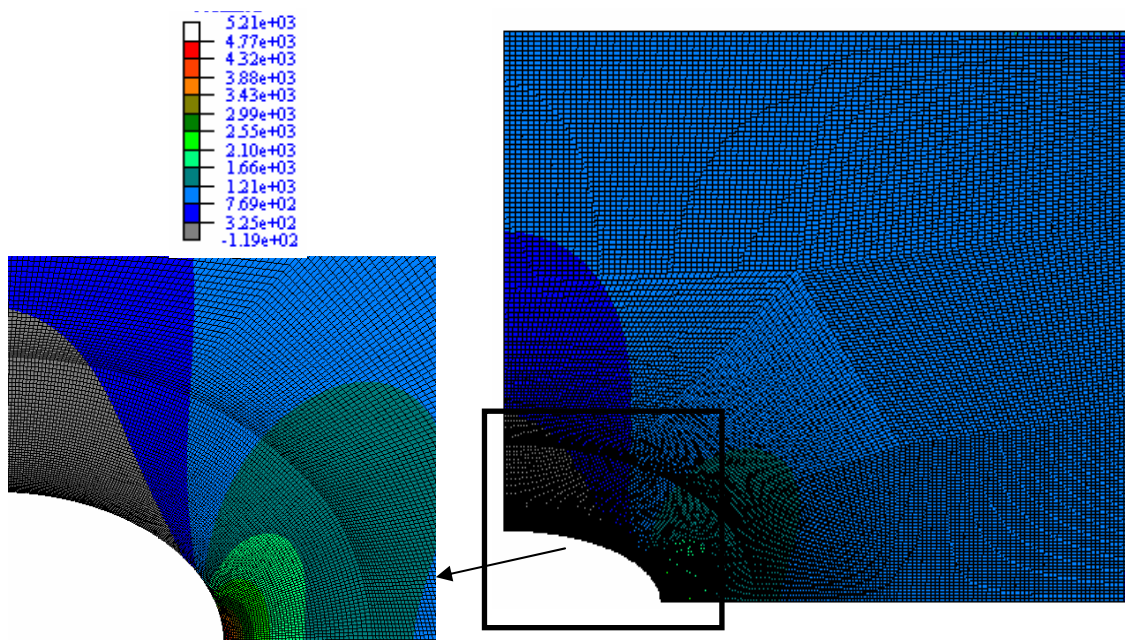


Fig. 46 Stress contours for σ_{22} for the reference solution.

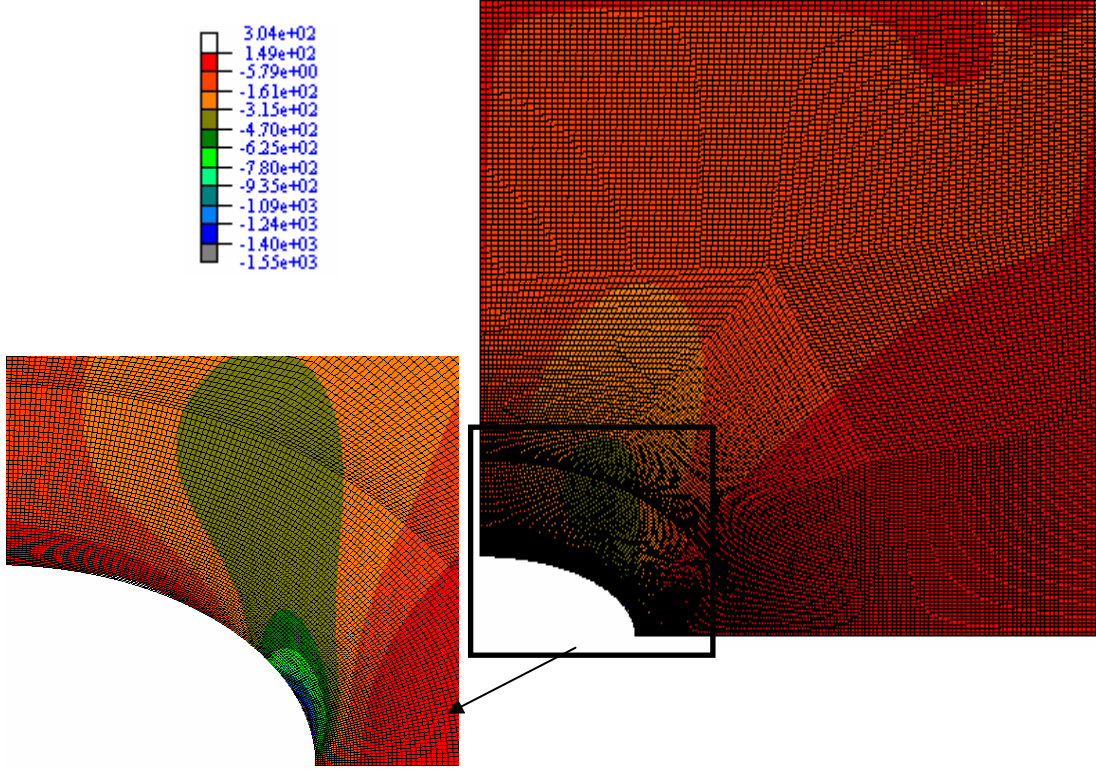


Fig. 47 Stress contours for σ_{12} for the reference solution.

Figure 45 shows the mesh considered for the reference solution. The figure also shows the deformed quarter plate along with the stress contours for the σ_{11} direction. The inset shows the stress distribution near the hole. Similar plots are shown in figure 46 and 47 for stress components σ_{22} and σ_{12} respectively.

Figure 48 shows the cluster mesh considered for the B-spline finite element analysis. This mesh was used to define the geometry of the problem. The mid-side nodes are biased towards the hole, for the clusters near the hole, to obtain a graded mesh. To obtain the conventional finite element mesh, the same super elements are further divided to obtain a finer graded mesh. Based on the number of degrees of freedom associated with each cluster, the number of degrees of freedom for the entire analysis is varied. Several cases are run for different number of degrees of freedom to study the convergence behavior for both the conventional and B-spline finite elements.

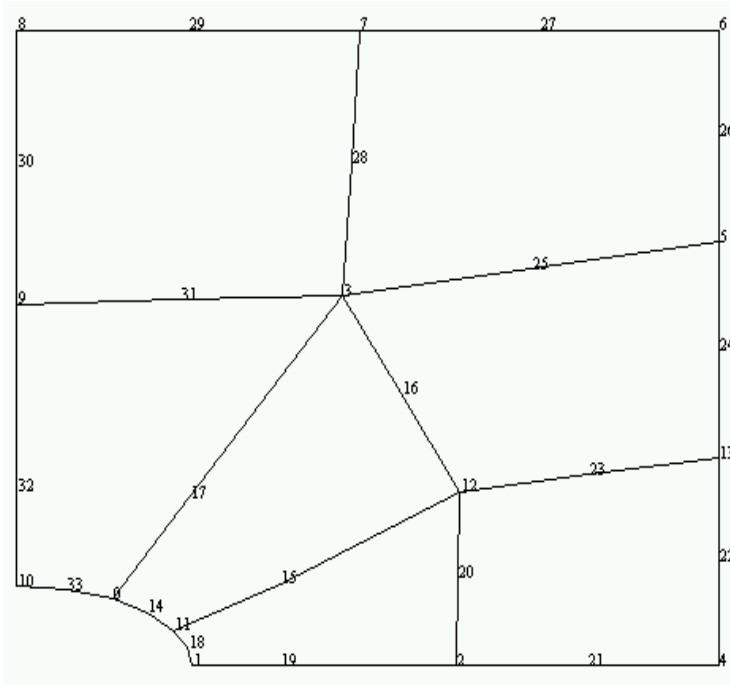


Fig. 48 Cluster mesh for B-spline finite element.

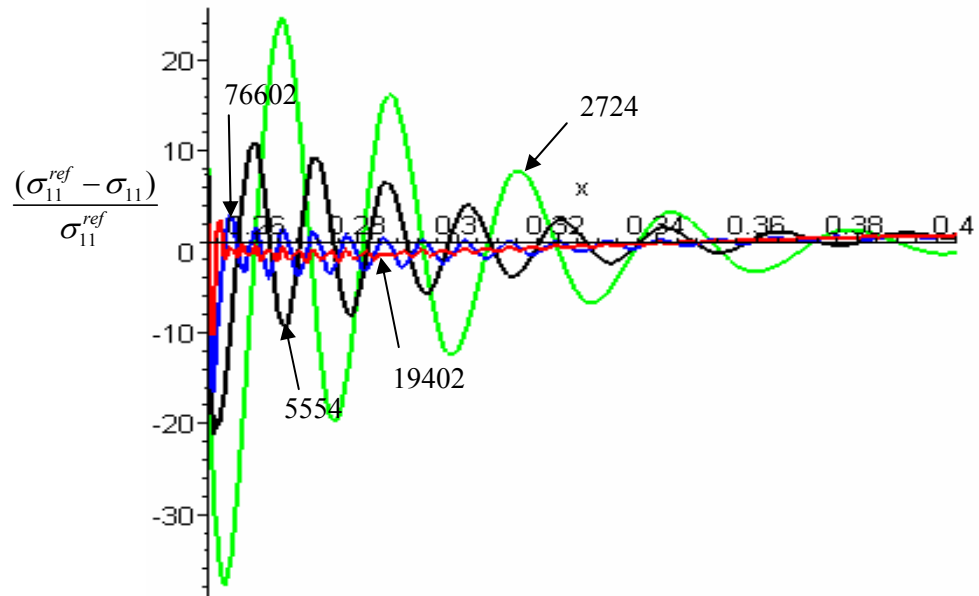


Fig. 49 Normalized difference in σ_{11} value with respect to the reference value for conventional finite elements.

The results for these cases are compared by evaluating the stress along the $y = 0$ line. The error in the stress values along the same line is calculated using the reference solution. Figure 49 shows the plot of the error in σ_{11} calculated for the conventional finite elements with respect to the reference solution for four different degrees of freedom values: 76602, 19402, 5554, 2724. The error shows lower oscillation with an increase in the number of degrees of freedom for the four different cases of degrees of freedom considered. Similarly, fig. 50 shows the plot if the error in σ_{11} using B-spline finite elements for degrees of freedom considered as 24524, 12004, 5964 and 1564. The change in oscillations is not as high as in the case of conventional elements, and the amplitude of oscillations is much lower than the conventional elements. The two plots show the error in the case of B-spline elements to be smaller and show a much higher convergence rate for the B-spline elements as compared to the conventional elements. These two plots show the convergence rate with respect to the number of degrees of freedom, and it can be seen that the convergence rate for B-spline is much faster than the conventional finite elements.

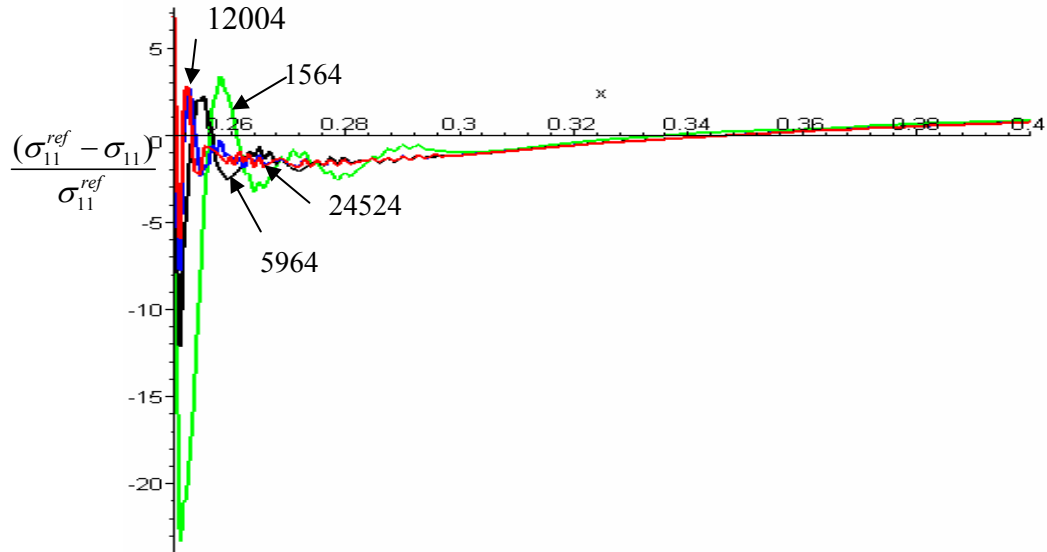


Fig. 50 Normalized difference in σ_{11} value with respect to the reference value for B-spline finite elements.

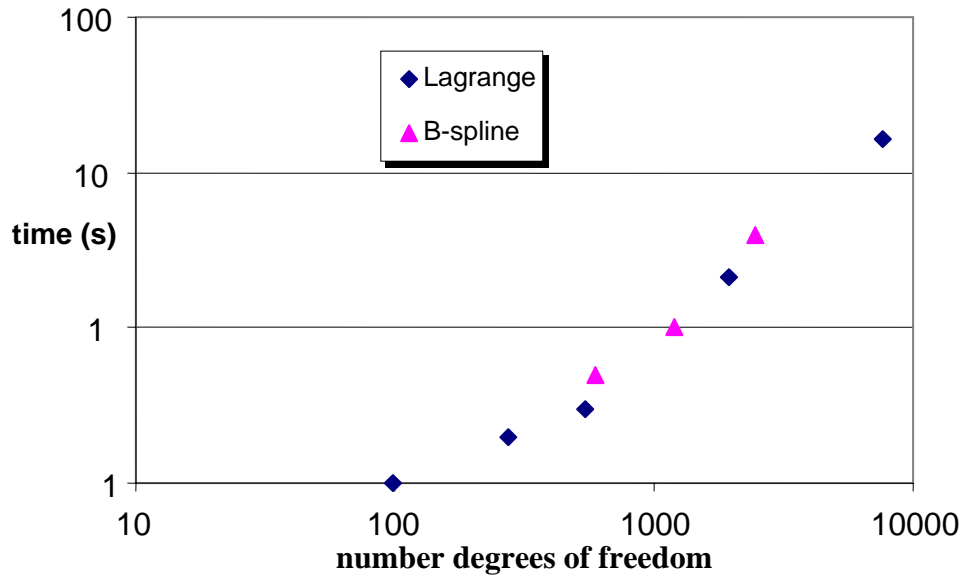


Fig. 51 Solution time vs number of degrees of freedom for convetional and B-Spline finite elements.

In order to compare the efficiency of the two methods, the time required to solve the two problems needs to be compared. Figure 51, shows a comparison of the time required to solve the equations for different number of degrees of freedom for both the cases. It can be seen from the plot that the time for B-spline is slightly higher than the Lagrange elements, which was expected. The error in the stress component σ_{11} for both the cases is compared against the solving time. The plot showing the error in σ_{11} along $y = 0$ line vs time is shown in fig. 52. For the lower number of degrees of freedom the time required to solve in the conventional case is less than the B-spline elements, but the error calculated is much larger than the same degree of freedom B-spline case. For the same error value, the time required to solve the equations for conventional finite elements is larger than for the B-spline elements.

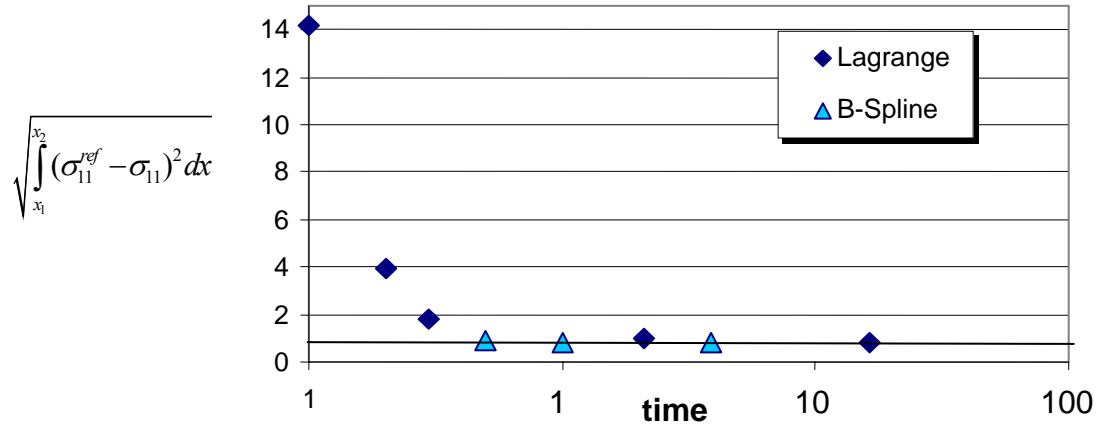


Fig. 52 Error in σ_{11} vs time for both conventional and B-spline finite elements.

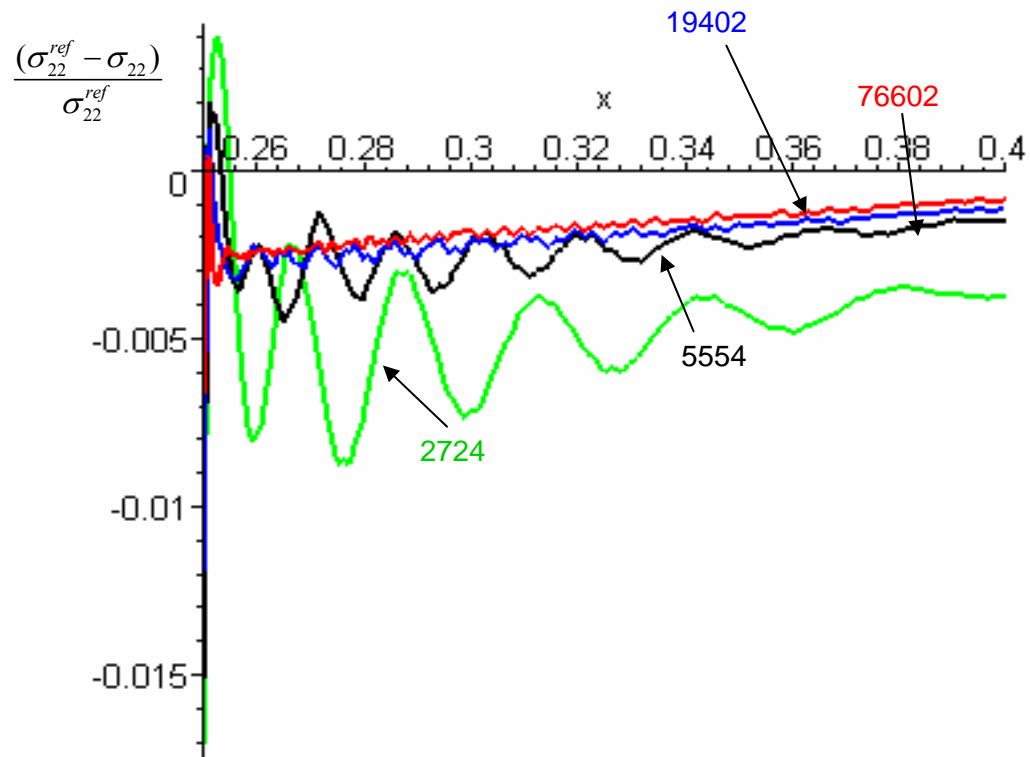


Fig. 53 Normalized difference in σ_{22} value with respect to the reference value for conventional finite elements.

A similar study is conducted for the stress component σ_{22} , which is the largest along the $y = 0$ line. The error in σ_{22} along the line is plotted in fig. 53 for the four different cases: 76602, 19402, 5554 and 2724 for conventional finite elements. The error for the B-spline finite elements for degrees of freedom 24524, 12004, 5964 and 1564 is shown in figure 54. Comparing the two plots, it can be observed that the curve showing the error for the case of 1564 B-spline finite element dof's is much less than even 2724 dof's for the conventional finite elements. The error for the 1564 dof B-spline case is almost the same as the conventional 5554 dof case. Also the four cases in the B-spline case only differ near the hole, and are almost overlapping for the entire length of the plate. The convergence behavior observed in this case for B-spline is again much better than the conventional finite elements.

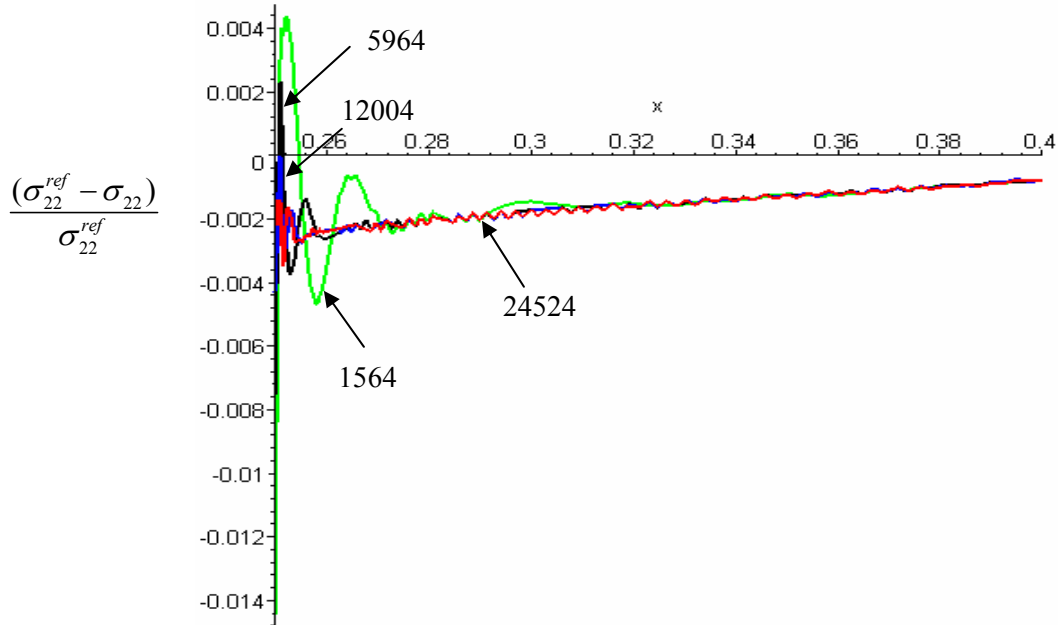


Fig. 54 Normalized difference in σ_{22} value with respect to the reference value for B-spline finite elements.

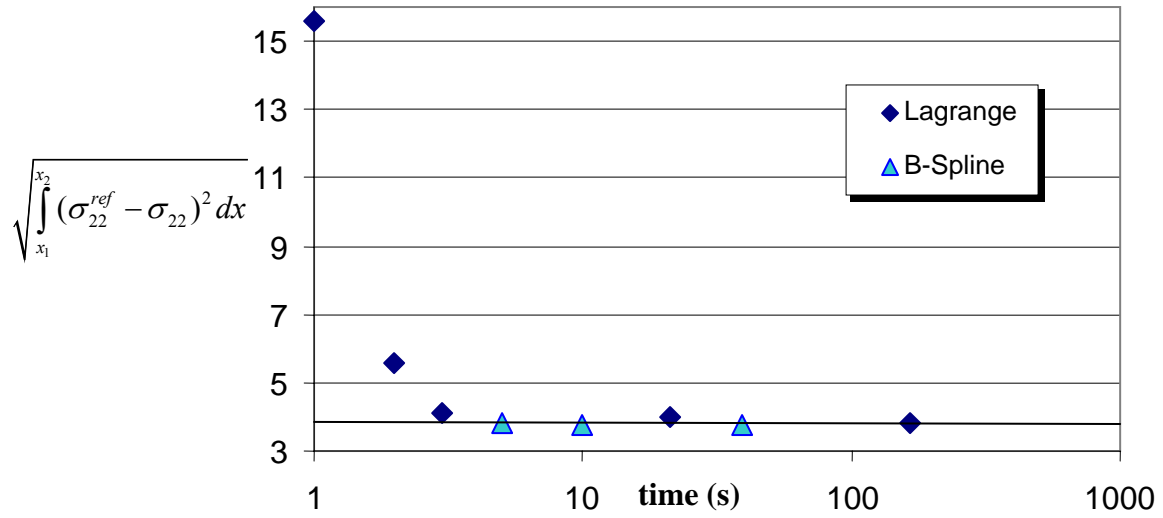


Fig. 55 Error in σ_{22} vs time for both conventional and B-spline finite elements.

The error vs solution time plot for the stress component σ_{22} is shown in fig. 55 for both the B-spline and conventional elements case. The trend shown is very similar to the σ_{11} case shown in fig. 52. The time required to reach the same level of accuracy for B-spline seems to be less than for the conventional elements. The error value reaches its lowest much faster, for a much lower number of degrees of freedom than the conventional finite element case.

CHAPTER VI

CONCLUSIONS AND FUTURE WORK

A novel method for finite element analysis for one and two dimensional problems is introduced in which B-spline interpolations are used instead of the Lagrange interpolations as in the case of conventional finite elements. The technique takes advantage of the higher continuity of the B-spline interpolation resulting in the development of a much efficient and accurate analysis tool. A strategy to implement the interpolation functions used in CAD modeling in the existing finite element model has been developed.

A convergence study and error analysis has been conducted to compare the efficiency of the performance of B-spline finite elements with the Lagrange interpolation used for finite elements. The error as compared to the number of degrees of freedom in the case of B-spline finite elements is less than the conventional method. The new technique is more efficient in terms of the time required to acquire the same level of accuracy as the conventional finite element analysis. This is particularly useful for cases where there are large stress gradients, and require a lot of run time for an accurate analysis. The performance and efficiency of the B-spline finite elements is considerably better than the conventional elements.

This method seems to have a huge potential for finite element problems requiring huge solution time, especially for the case of three dimensional finite elements. The program can be extended for the complex three dimensional problems which can result in a much more efficient analysis. More robust techniques for grading the mesh need to be developed for a more efficient analysis. Adaptive refinement strategies need to be developed for optimization.

The accuracy and efficiency for the two methods is analyzed by only calculating the error in stress distribution along a line. Some more comprehensive error analysis techniques for the overall model need to be developed and compared. Also, the

efficiency of the two methods needs to be compared for elements which use selective reduced integration to improve performance.

The design process involves two main steps, the design using Computer Aided Design (CAD) tools and then further analysis using Computer Aided Engineering (CAE) tools. The use of CAE requires remeshing of the model generated from CAD, since the interpolation functions used in both cases are different. The CAD models use B-spline interpolation whereas the CAE uses Lagrange type interpolation. The use of B-spline interpolation for CAE can help in integrating the two steps. The complexity involved in the generation of a new mesh for CAE can be eliminated by the use of this method, but there needs to be future work done to be able to use the data generated by CAD to be automatically used by CAE. This would eliminate the cumbersome process of meshing, and would make the design process more optimized.

REFERENCES

1. Turner, M. J., Clough, R. W., Martin, H. C., and Topp, L. J., (1956), Stiffness and Deflection Analysis of Complex Structures, *Journal of Aeronautical Sciences*, **23**, (9), 805-824.
2. Zienkiewicz, O.C., (1967), *The Finite Element Method in Structural and Continuum Mechanics*, McGraw-Hill, New York.
3. Rogers, D.F., (2001), *An Introduction to NURBS, with Historical Perspective*, Morgan Kaufmann Publishers, San Francisco.
4. Prenter, P. M., (1975), *Splines and Variational Methods*, Wiley, New York.
5. Hollig, K., (2003), *Finite Element Methods with B-Splines*, Frontiers in Applied Mathematics 26, SIAM, Philadelphia.
6. Sabin, M.A., (1997), Spline Finite Elements, PHd Dissertation, Cambridge University, U.K.
7. Schoenberg, I. J., (1946), Contributions to the Problem of Approximation of Equidistant Data by Analytic Functions, *Quart. Appl. Math.*, **4**, 45-99, 112-141.
8. Ahlberg, J. H., Nilson, E.N., and Walsh, J.L., (1967), *The Theory of Splines and Their applications*, Academic Press, New York.
9. De Boor, C., (1962), Bicubic spline interpolation, *J. Math. Phys.*, **41**, 212–218.
10. De Boor, C., (1978), *A Practical Guide to Splines*, Springer-Verlag, New York.
11. Cote, K., Complex 3D Flight Trajectory Generation and Tracking Using Cubic Splines”, *AIAA's 1st Technical Conference and Workshop on Unmanned Aerospace Vehicles*, 20-23 May 2002, Portsmouth, Virginia.

12. Zamani, N.G.,(1981), A Least Squares Finite Element Method Applied to B-Splines”, *Journal of the Franklin Institute*, **311**, (3), 195-208.
13. Zamani, N.G., (1981), Least Squares Finite Element Approximation of Navier’s Equation, *Journal of the Franklin Institute*, **311**, (5), 311-321.
14. Peng-Cheng, S. and Jian-Guo,W., (1987), Vibration Analysis of Flat Shells by using B Spline Functions, *Computers & Structures*, **25**, (1), 1-10.
15. Gupta, A., Kiusalaas, J. and Saraph, M., (1991), Cubic B-Spline for Finite Element Analysis of Axisymmetric Shells, *Computers & Structures*, **38**, (4), 463-468.
16. Gutkowski,R.M., (1991), Plate Bending Analysis by Unequally Spaced Splines, *Thin Walled Structures*, **11**, 409-430.
17. Benjeddou, A., (2000), Vibrations of Complex Shells of Revolution Using B-spline Finite Elements, *Computers & Structures*, **74**, 429-440.
18. Roh, H.Y., and Cho, M., (2004), The Application of Geometrically Shell Elements to B-Spline Surfaces, *Computer Methods in Applied Mechanics and Engineering*, **193**, 2261-2299.
19. Braibant,V. and Fleury,C., (1984), Shape Optimal Design Using B-Splines, *Computer Methods in Applied Mechanics and Engineering*, **44**, 247-267.
20. Schramm, U. and Pilkey, W. D., (1993), The Coupling of Geometric Descriptions and Finite Elements Using NURBs – A Study in Shape Optimization, *Finite Elements in Analysis and Design*, **15**, 11-34.

21. Onwubolu, G.C., (1989), Finite Element Mesh Generation of 3D-Surfaces in CAD, *Computers & Structures*, **32**, (1), 31-36.
22. Onwubolu, G.C., (1991), Finite Element Mesh Generation of Arbitrary Solid Objects in CAD, *Computers & Structures*, **41**, (5), 1011-1018.
23. Kagan, P., Fischer, A. and Bar-Yoseph, P.Z., (1998), New B-Spline Finite Element Approach for Geometrical Design and Mechanical Analysis, *International Journal for Numerical Methods in Engineering*, **41**, 435-458.
24. Kagan, P. and Fischer, A., (2000), Integrated Mechanically Based CAE System Using B-Spline Finite Elements, *Computer -Aided Design*, **32**, 539-552.
25. Roh, H.Y., and Cho, M., (2005), Integration of Geometric Design and Mechanical Analysis Using B-Spline Functions on Surface, *International Journal for Numerical Methods in Engineering*, **62**, 1927–1949.
26. Natekar, D., Zhang, X., and Subbarayan, G., (2004), Constructive Solid Analysis: A Hierarchical, Geometry-Based Meshless Analysis Procedure for Integrated Design and Analysis, *Computer-Aided Design*, **36**, 473-486.
27. Hughes, T.J.R., Cottrell, J.A., and Bazilevs, Y., (2005), Isogeometric Analysis: CAD, Finite Elements, NURBS, Exact Geometry and Mesh Refinement, *Computer Methods in Applied Mechanics and Engineering*, **194**, 4135-4195.
28. Poschl, W., (1998), B-Spline Finite Elements and their Efficiency in Solving Relativistic Mean Field Equations—*Computer Physics Communications*, **112**, 42-66.

29. Bahadir, A. R., (2004), Application of Cubic B-spline Finite Element Technique to the Thermistor Problem, *Applied Mathematics and Computation*, **149**, 379-387.
30. Kutluay, S. and Esen, A., (2004), A B-spline Finite Element Method for the Thermistor Problem with the Modified Electrical Conductivity, *Applied Mathematics and Computation*, **156**, 621-632.
31. Kutluay, S., Esen, A., and Dag, I., (2004), Numerical Solutions of the Burgers' Equation by the Least-Squares Quadratic B-spline Finite Element Method, *Journal of Computational and Applied Mathematics*, **167**, 21-33.
32. Dag, I., Irk, D., and Saka, B., (2005), A Numerical Solution of the Burger's Equation Using Cubic B-splines, *Applied Mathematics and Computation*, **163**, 199-211.
33. Aksan, E.N., (2006), Quadratic B-spline Element Method for Numerical Solution of the Burgers' Equation, *Applied Mathematics and Computation*, **174**, 884-896.
34. Lagrange, J.L., *Leçons élémentaires sur les mathématiques, données à l'Ecole Normale en 1795*, in Oeuvres VII, Gauthier-Villars, Paris, 1877, pp. 183-287.
35. Reddy, J.N., (1984), *An Introduction to the Finite Element Method*, McGraw-Hill, New York.

VITA

BACKGROUND

Name : Bhavya Aggarwal
Address : C – 2/164, West Enclave, Pitampura, New Delhi – 110034, INDIA
Phone : 91-120-4336319
Email : abhavya@gmail.com

EDUCATIONAL BACKGROUND

- Bachelor of Engineering
Punjab Engineering College,
Panjab University, INDIA
Graduation : Spring 2002
- Master of Science
Texas A&M University, USA
Graduation : December 2006